

**APLIKASI *MOBILE* PENCARIAN RUTE TERPENDEK
LOKASI FASILITAS UMUM BERBASIS ANDROID
MENGUNAKAN ALGORITMA *FLOYD-WARSHALL***

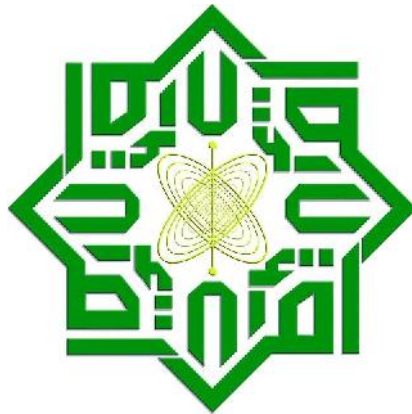
TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar
Sarjana Teknik Pada Jurusan Teknik Informatika

Oleh:

RIZKY YUSAPUTRA

10751000282



**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
PEKANBARU**

2013

APLIKASI *MOBILE* PENCARIAN RUTE TERPENDEK LOKASI FASILITAS UMUM BERBASIS ANDROID MENGUNAKAN ALGORITMA *FLOYD-WARSHALL*

RIZKY YUSAPUTRA

10751000282

Tanggal Sidang: 17 Januari 2013

Periode Wisuda : Februari 2013

Jurusan Teknik Informatika

Fakultas Sains dan Teknologi

Universitas Islam Negeri Sultan Syarif Kasim Riau

ABSTRAK

Pencarian suatu lokasi dengan rute terpendek merupakan suatu cara untuk membantu pengguna dalam mendapatkan jalur terdekat. Salah satu algoritma dalam melakukan pencarian rute terpendek adalah algoritma *Floyd-Warshall*. Algoritma ini melakukan pemecahan masalah dengan memandang solusi akhir yang akan diperoleh sebagai suatu keputusan yang saling terkait. Pada penelitian ini dalam melakukan pencarian rute terpendek *database* yang digunakan hanya *database* jalan utama, arteri dan lokasi fasilitas umum saja. Data jalan dan lokasi fasilitas umum yang digunakan untuk pengujian diambil dari situs *navigasi.net* yaitu website yang menyediakan *Point of Interest* (POI) terkait lokasi-lokasi tujuan yang ada di Indonesia. Pengujian yang akan dilakukan meliputi pengujian dari aspek jarak dan waktu yang dapat ditempuh. Hasil dari sistem pencarian rute terpendek yang akan dibangun adalah *direction* dengan rute terpendek, total jarak dan waktu yang akan di tempuh. Dan berdasarkan hasil pengujian dalam melakukan pencarian rute terpendek di 15 lokasi tujuan yang berbeda dapat disimpulkan bahwa algoritma *Floyd-Warshall* ini sudah dapat menghasilkan solusi yang lebih optimum dari pada algoritma pencarian rute terpendek yang lainnya, akan tetapi untuk waktu yang dibutuhkan dalam melakukan proses pencarian tersebut memang relatif lebih lama dari algoritma yang lain.

Kata kunci: Algoritma *Floyd-Warshall*, *Android Mobile*, Fasilitas Umum, Rute Terpendek.

DAFTAR ISI

HALAMAN JUDUL LAPORAN	i
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL	iv
LEMBAR PERNYATAAN	v
LEMBAR PERSEMBAHAN	vi
ABSTRAK	vii
<i>ABSTRACT</i>	viii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xiv
DAFTAR TABEL	xv
DAFTAR ALGORITMA	xvi
DAFTAR LAMPIRAN	xvii
DAFTAR SIMBOL	xviii
DAFTAR SINGKATAN	xix
BAB I PENDAHULUAN.....	I-1
1.1. Latar Belakang	I-1
1.2. Rumusan Masalah	I-3
1.3. Batasan Masalah.....	I-3
1.4. Tujuan Penelitian	I-4
1.5. Sistematika Penulisan	I-4
BAB II LANDASAN TEORI	II-1
2.1. Pencarian Rute Terpendek	II-1
2.2. Pengenalan Algoritma Rute Terpendek	II-2
2.2.1. Algoritma <i>Floyd-Warshall</i>	II-3
2.2.1.1. Deskripsi Algoritma	II-4

2.2.1.2. Karakteristik Algoritma	II-5
2.2.1.2. Pemecahan Masalah Algoritma	II-5
2.3. <i>Global Positioning System</i> (GPS)	II-6
2.3.1. Keunggulan GPS	II-8
2.3.2. GPS dan A-GPS	II-9
A. <i>Global Positioning System</i> (GPS)	II-10
B. <i>Assisted- Global Positioning System</i> (A-GPS)	II-10
2.4. Android	II-11
2.4.1. Arsitektur Android	II-12
2.5. Analisa dan Perancangan Berorientasi Objek	II-14
2.5.1. <i>Unified Modelling Language</i> (UML)	II-15
2.5.2. <i>Use Case Diagram</i>	II-15
2.5.3. <i>Class Diagram</i>	II-15
2.5.4. <i>Sequence Diagram</i>	II-16
2.6. <i>Rational Unified Process</i> (RUP)	II-16
2.6.1. Pengertian RUP	II-16
2.6.2. Fase RUP	II-18
2.7. Sistem Informasi Geografis (SIG)	II-20
2.7.1. Pengertian SIG	II-20
2.7.2. Subsistem SIG	II-22
2.7.3. Konsep Model Data Spasial SIG	II-23
2.7.4. <i>Universal Transverse Mercator</i> (UTM)	II-24
2.8. Rumus Perhitungan Waktu Tempuh	II-24
BAB III METODOLOGI PENELITIAN	III-1
3.1. Alur Tahapan RUP	III-1
3.1.1. Fase <i>Inception</i>	III-2
3.1.2. Fase <i>Elaboration</i>	III-3
3.1.3. Fase <i>Construction</i>	III-3
3.1.4. Fase <i>Transition</i>	III-3
BAB IV ANALISA DAN PERANCANGAN	IV-1
4.1. Gambaran Umum Sistem	IV-1

4.2. Algoritma <i>Floyd-Warshall</i>	IV-2
4.2.1. Cara Kerja Algoritma <i>Floyd-Warshall</i>	IV-3
4.2.2. Perhitungan Manual Algoritma <i>Floyd-Warshall</i>	IV-4
4.2.3. Perhitungan Waktu Tempuh	IV-9
4.3. Fungsi Sistem	IV-9
4.3.1. Fungsi Sistem dari Sisi Perangkat Android	IV-9
4.3.2. Fungsi Media Penghubung	IV-9
4.4. Deskripsi Pengguna	IV-10
4.5. Perancangan Sistem	IV-10
4.5.1. Model <i>Use Case</i>	IV-10
4.5.2. <i>Class Diagram</i>	IV-11
4.5.3. <i>Sequence Diagram</i>	IV-13
4.5.4. <i>Activity Diagram</i>	IV-14
4.6. Perancangan Interface	IV-14
BAB V IMPLEMENTASI DAN PENGUJIAN.....	V-1
5.1. Implementasi	V-1
5.1.1. Lingkungan Pengembangan	V-1
5.1.3. Lingkungan Implementasi	V-2
5.1.4. Tahap-Tahap Implementasi	V-2
5.2. Pengujian <i>Blackbox</i> Aplikasi <i>Pekanbaru mGuide</i>	V-4
5.3. Pengujian Akses Aplikasi <i>Pekanbaru mGuide</i>	V-5
5.4. Kesimpulan Pengujian	V-13
BAB VI PENUTUP	VI-1
6.1. Kesimpulan.....	VI-1
6.2. Saran.....	VI-2
DAFTAR PUSTAKA	xx
LAMPIRAN	
DAFTAR RIWAYAT HIDUP	

DAFTAR GAMBAR

Gambar	Halaman
2.1. Contoh Pencarian Rute Terpendek	II-2
2.2. Contoh-Contoh GPS	II-7
2.3. Penentuan Posisi GPS dan A-GPS	II-11
2.4. Arsitektur Android	II-12
2.5. Struktur Proses 2 Dimensi RUP	II-17
2.6. Komponen Sistem Informasi Geografis	II-21
2.7. Subsistem Sistem Informasi Geografi	II-23
3.1. Tahapan Penelitian	IV-1
4.1. Gambaran Umum Sistem	IV-2
4.2. Contoh Node Jalan Berbobot	IV-4
4.3. <i>Flowchart</i> Perhitungan Algoritma <i>Flowd-Warshall</i>	IV-6
4.4. Tahap 1	IV-7
4.5. Tahap 2	IV-8
4.6. <i>Use Case Diagram</i>	IV-25
4.7. <i>Class Diagram</i> <i>PKU mGuide</i>	IV-26
4.8. <i>Sequence Diagram</i> <i>Search</i>	IV-27
4.9. <i>Activity Diagram</i> Buka Aplikasi	IV-28
4.10. Tampilan Utama <i>Pekanbaru mGuide</i>	IV-29
5.1. Hasil Implementasi Menampilkan Menu <i>Home</i>	V-3

DAFTAR TABEL

Tabel	Halaman
4.1. Deskripsi Pengguna	IV-10
4.2. Spesifikasi <i>Use Case Diagram</i>	IV-11
4.3. Deskripsi Perancangan <i>Class Diagram</i>	IV-12
5.1. Pengujian Aplikasi dengan Metode <i>Blackbox</i>	V-4
5.2. Pengujian Akses ke Aplikasi dari Device Android (Andromax-i)	V-5
5.3. Pengujian Aplikasi <i>Pekanbaru mGuide</i> dari Beberapa Device	V-11

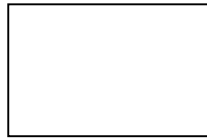
DAFTAR ALGORITMA

Algoritma	Halaman
2.1. <i>Pseudo Code Floyd-Warshall</i>	II-4
2.2. Pemecahan Solusi Terpendek	II-6
5.1. koneksi.php	V-3

DAFTAR LAMPIRAN

Lampiran	Halaman
A. Perancangan Model Sistem	A-1
B. Deskripsi Perancangan Antar Muka.....	B-1
C. <i>Source Code</i> Penghubung	C-1
D. Hasil Implementasi	D-1
E. Perhitungan Manual Algoritma <i>Floyd-Warshall</i>	E-1

DAFTAR SIMBOL



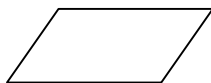
Proses pada *flowchart*



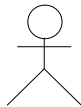
Start/ Finish suatu proses pada *flowchart*



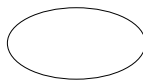
Alur/ langkah pada *flowchart* dan model data spasial



Input/ Output pada *flowchart*



Pengguna Sistem (Aktor)



Proses (*Use Case*)

DAFTAR SINGKATAN

LBS	: <i>Location Based Service</i>
GPS	: <i>Global Positioning System</i>
DVM	: <i>Dalvik Virtual Machine</i>
UML	: <i>Unified Modelling Language</i>
OOAD	: <i>Object Oriented Analysis Design</i>
RUP	: <i>Rational Unified Process</i>
UTM	: <i>Universal Transverse Mercator</i>

BAB I

PENDAHULUAN

1.2. Latar Belakang

Pekanbaru adalah ibukota Provinsi Riau, merupakan pusat perdagangan, bisnis, industri, pendidikan, dan jasa di kawasan Sumatera. Pekanbaru juga merupakan kota dengan pertumbuhan populasi penduduk dan perkembangan ekonomi tertinggi di Indonesia (Tribunnews, 2011). Kota Pekanbaru memiliki sungai Siak yang menjadi jalur strategis bagi kapal-kapal yang akan menuju Singapura, Malaysia dan daerah lain di Riau serta provinsi lain di Sumatera. Kota Pekanbaru diproyeksikan menjadi kota jasa, sehingga Kota Pekanbaru saat ini terus berupaya membenahi diri dengan meningkatkan fasilitas-fasilitas penunjang perkotaan dan fasilitas-fasilitas umum yang lengkap.

Kini Kota Pekanbaru yang berperan sebagai kota bisnis dan jasa telah menjadi daya tarik tersendiri bagi masyarakat lokal maupun internasional. Dengan seiring berjalannya waktu jumlah penduduk Kota Pekanbaru semakin bertambah. Pada tahun 2008 saja tercatat ada 799.213 jiwa, tahun 2009 meningkat menjadi 802.788 jiwa dan pada tahun 2010 tercatat ada 897.768 jiwa. (<http://bappeda.pekanbaru.go.id>, 2012).

Dengan semakin bertambahnya jumlah penduduk dari tahun ke tahun maka, semakin banyak pula masyarakat yang membutuhkan informasi tentang rute terpendek mana yang harus dilewati masyarakat untuk dapat mencapai lokasi-lokasi fasilitas yang ditentukan di Kota Pekanbaru. Karena kebanyakan masyarakat Pekanbaru sudah mengetahui lokasi-lokasi fasilitas umum yang ada di Kota Pekanbaru, akan tetapi tidak banyak yang mengerti rute mana yang terpendek untuk mencapai lokasi fasilitas-fasilitas umum tersebut, dan hal inilah yang menyebabkan masyarakat mengalami kebingungan.

Berdasarkan masalah diatas maka dilakukan penelitian tugas akhir yang telah diberikan perhitungan untuk dapat menghasilkan solusi rute terpendek menjadi lebih optimal yaitu dengan perhitungan Algoritma *Floyd-Warshall*.

Dengan algoritma ini, pengguna dapat mengetahui rute-rute terpendek untuk menuju ke tempat-tempat fasilitas umum lebih mudah.

Penelitian tentang pencarian lokasi fasilitas umum di Kota Pekanbaru sendiri sudah pernah dilakukan oleh Dian Fitriyani, yang merancang suatu Sistem Informasi *Location Based Service* (LBS) berbasis *mobile* dengan teknologi J2ME untuk pencarian lokasi fasilitas umum yang ada di Kota Pekanbaru, sistem ini tidak menggunakan metode atau algoritma apapun. Sistem tersebut hanya menyajikan informasi fasilitas umum yang berbasis lokasi, maksudnya sistem hanya menampilkan lokasi fasilitas umum saja, tidak menampilkan rute mana yang harus di lewati untuk mencapai fasilitas yang dibutuhkan (Fitriyani, 2011).

Sebenarnya ada banyak algoritma yang dapat menyelesaikan masalah pencarian rute terpendek, antara lain Algoritma *Dijkstra*, Algoritma *Floyd-Warshall*, Algoritma *Bellman-Ford*, dan lain-lain. Akan tetapi merujuk pada jurnal/penelitian yang dilakukan oleh Raden Aprian Diaz Novandi dimana, kesimpulan yang dihasilkan adalah: **“Algoritma Floyd-Warshall yang menerapkan pemrograman dinamis lebih menjamin keberhasilan penemuan solusi optimum untuk kasus penentuan lintasan terpendek (*all-pairs shortest path*)”**. (Novandi, 2007)

Dari penelitian diatas dapat dilihat bahwa, algoritma-algoritma ini memiliki kelebihan dan kekurangannya masing-masing, inilah yang menjadi dasar dan alasan mengapa Algoritma *Floyd-Warshall* yang digunakan untuk membantu memecahkan masalah pada tugas akhir ini, karena Algoritma *Floyd-Warshall* lebih memandang solusi akhir yang akan diperoleh sebagai sesuatu keputusan yang saling terkait. Artinya solusi-solusi tersebut dibentuk dari solusi yang berasal dari tahap sebelumnya. Sehingga dengan memadukan antara *mobile* yang berbasiskan Android dan Algoritma *Floyd-Warshall* diharapkan dapat mempermudah pengguna dalam melakukan pencarian lokasi fasilitas umum yang dibutuhkan dengan jarak yang terdekat dan dapat memberikan keputusan yang lebih akurat.

Berdasarkan latar belakang diatas, maka dilakukanlah perancangan aplikasi *mobile* berbasis Android yang berfungsi untuk mencari lokasi fasilitas

umum dengan rute terpendek menggunakan Algoritma *Floyd-Warshall*. Sehingga nantinya selain pengguna dapat melihat posisinya saat itu dan posisi beberapa fasilitas umum yang ada, pengguna juga dapat mengetahui rute mana yang harus mereka lewati untuk mencapai lokasi masing-masing fasilitas umum yang ada di Kota Pekanbaru dengan cara yang lebih mudah dan lebih cepat.

1.3. Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka pokok permasalahan dipersempit. Permasalahannya adalah “bagaimana merancang dan membangun suatu aplikasi *mobile* berbasis Android yang dapat mencari lokasi fasilitas umum dengan rute terpendek menggunakan Algoritma *Floyd-Warshall*”

1.4. Batasan Masalah

Batasan masalah yang terdapat pada penelitian ini adalah sebagai berikut:

1. Peta yang ditampilkan dalam bentuk vektor, yaitu format titik, garis, dan polygon.
2. Digitasi (pemetaan) yang dilakukan dibatasi hanya pada jalan arteri dan jalan utama.
3. Penyajian informasi dibatasi pada fasilitas umum berupa sajian informasi mengenai fasilitas-fasilitas umum yang ada di Kota Pekanbaru yaitu ATM Tunai (BNI, Mandiri dan BCA), Stasiun Pengisian Bahan Bakar Umum (SPBU), masjid, rumah makan, hotel, rumah sakit, bandara, dan terminal.
4. Perhitungan waktu tempuh dengan kecepatan rata-rata 40 km/h dan tidak memperhitungkan hambatan-hambatan yang ada.

1.5. Tujuan Penelitian

Adapun tujuan dari penelitian tentang Sistem Informasi Geografis ini adalah merancang bangun aplikasi *mobile* Android yang bertujuan untuk mempermudah pengguna dalam melakukan pencarian lokasi fasilitas umum menggunakan rute terpendek dan rute mana yang harus mereka lalui untuk mencapai fasilitas umum yang dituju.

1.6. Sistematika Penulisan

Sistematika penulisan dari skripsi ini terdiri dari beberapa bagian utama sebagai berikut:

BAB I Pendahuluan

Dalam bab ini diuraikan tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian, dan sistematika penulisan skripsi.

BAB II Landasan Teori

Dalam bab ini diuraikan landasan teori yang digunakan dalam memecahkan masalah dan membahas masalah yang ada. Bab ini membahas teori-teori yang berkaitan dengan Android, Sistem Informasi Geografis, GPS (*Global Positioning System*) dan Algoritma *Floyd-Warshall*.

BAB III Metodologi Penelitian

Bab ini membahas langkah-langkah yang dilaksanakan dalam proses penelitian, yaitu pengamatan pendahuluan dan pengumpulan data, tahapan identifikasi masalah, perumusan masalah, analisa aplikasi, perancangan aplikasi dan implementasi beserta pengujian.

BAB IV Analisis Dan Perancangan Sistem

Dalam bab ini diuraikan tentang analisis kebutuhan perangkat lunak yang akan dikembangkan, beserta perancangan.

BAB V Implementasi Dan Pengujian

Dalam bab ini diuraikan tentang implementasi dan pengujian dari sistem yang dibangun berdasarkan hasil analisa dan perancangan pada bab sebelumnya.

BAB VI Penutup

Dalam bab ini diuraikan tentang saran dan kesimpulan yang didapat setelah pelaksanaan skripsi ini, beserta saran-saran untuk perbaikan pada pengembangan berikutnya.

BAB II

LANDASAN TEORI

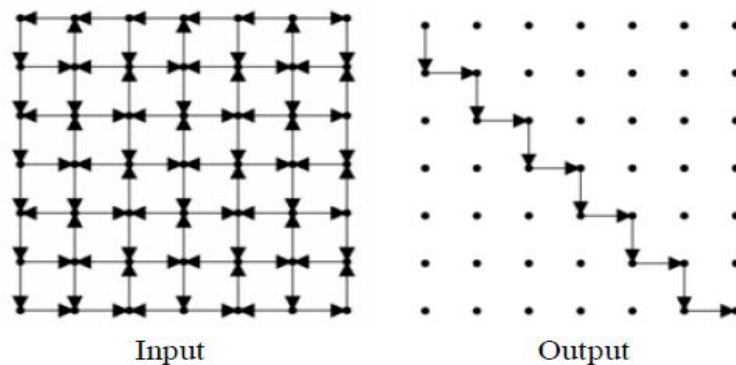
2.2. Pencarian Rute Terpendek

Lintasan terpendek adalah lintasan minimum yang diperlukan untuk mencapai suatu tempat dari tempat tertentu. Lintasan minimum yang dimaksud dapat dicari dengan menggunakan graf. Graf adalah sekumpulan titik di dalam bidang dua dimensi yang dihubungkan dengan sekumpulan garis (*edge*). Sebuah graf dibentuk dari kumpulan titik yang dihubungkan dengan garis-garis. Ada beberapa istilah yang berhubungan dengan graph, antara lain (Taurus, 2012):

1. Titik – titik tersebut disebut *vertex*.
2. Garis – garis yang menghubungkan antar *vertex* disebut *edge*.
3. *Adjacent* artinya bertetangga. Maksudnya jika ada dua *vertex* disebut *adjacent*, jika mempunyai *edge* yang sama.
4. Adalah bobot yang biasanya terdapat pada *edge* yang merepresentasikan jarak dari *vertex-vertex* yang dihubungkan oleh *edge* tersebut.
5. *Path* adalah lintasan yang melalui *edge* dan *vertex* dalam graf.
6. *Cycle* adalah lintasan yang dimulai dan berakhir pada *vertex* yang sama.
7. *Direct* pada *directed graph* adalah graf dimana *edge-edgenya* mempunyai suatu arah.

Graf yang digunakan adalah graf-graf yang berbobot, yaitu graf yang setiap sisinya diberikan suatu nilai. Ada beberapa macam persoalan lintasan terpendek, antara lain (Arsa, 2012):

1. Lintasan terpendek antara dua buah simpul tertentu. (*a pair shortest path*).
2. Lintasan terpendek antara semua pasangan simpul (*all pairs shortest path*).
3. Lintasan terpendek dari simpul tertentu ke semua simpul yang lain (*single-source shortest path*).
4. Lintasan terpendek antara dua buah simpul yang melalui beberapa simpul tertentu (*intermediate shortest path*).



Gambar 2.1. Contoh Pencarian Rute Terpendek

2.3. Pengenalan Algoritma Pencarian Rute Terpendek

Algoritma adalah kumpulan perintah yang dibuat secara jelas dan sistematis berdasarkan urutan yang logis (logika) untuk penyelesaian suatu masalah. Namun algoritma pencarian rute tujuannya adalah algoritma yang menentukan bagaimana memilih rute optimal antara awal dan tujuan dengan memperhitungkan waktu kalkulasi terpendek. Ada beberapa Algoritma yang sudah dikembangkan, antara lain Algoritma *Dijkstra*, Algoritma *Floyd-Warshall*, Algoritma *Bellman-Ford*, Algoritma *Ant*, Algoritma *A**, dan lain-lain. Dimana inti logika dari algoritma-algoritma tersebut adalah sama, yaitu menentukan jarak terpendek dari setiap titik yang telah dibangun (Siswanto, 2010).

Penelitian yang dilakukan oleh Apri Kamayudi adalah membandingkan antara Algoritma *Bellman-Ford*, Algoritma *Dijkstra* dan Algoritma *Floyd-Warshall*. Dari hasil penelitian tersebut, kesimpulan yang dihasilkan adalah: (Kamayudi, 2006)

1. Urutan algoritma penyelesaian persoalan lintasan terpendek berdasarkan kompleksitas algoritmanya adalah:

$$\textit{Bellman-Ford} < \textit{Dijkstra} < \textit{Floyd-Warshall}$$

2. Berdasarkan masalah yang dapat diselesaikan:

Dijkstra => untuk masalah *single-source shortest path*

Bellman-Ford => untuk masalah *single-source shortest path*

Floyd-Warshall => untuk masalah *all-pairs shortest path*

Dari penelitian diatas dapat dilihat bahwa, algoritma-algoritma ini memiliki kelebihan dan kekurangannya masing-masing dalam menyelesaikan persoalan lintasan terpendek, karena masing-masing algoritma memiliki spesifikasi penyelesaian masalah, kompleksitas, waktu penyelesaian, serta jenis masalah yang berbeda, dan yang menjadi dasar serta alasan mengapa Algoritma *Floyd-Warshall* yang digunakan untuk membantu memecahkan masalah pada tugas akhir ini, karena Algoritma *Floyd-Warshall* lebih memandang solusi akhir yang akan diperoleh sebagai sesuatu keputusan yang saling terkait. Artinya solusi-solusi tersebut dibentuk dari solusi yang berasal dari tahap sebelumnya

2.3.1. Algoritma *Floyd-Warshall*

Algoritma *Floyd-Warshall* adalah salah satu varian dari pemrograman dinamis, yaitu suatu metode yang melakukan pemecahan masalah dengan memandang solusi yang akan diperoleh sebagai suatu keputusan yang saling terkait. Artinya solusi-solusi tersebut dibentuk dari solusi yang berasal dari tahap sebelumnya. Algoritma ini menghitung bobot terkecil dari semua jalur yang menghubungkan sebuah pasangan titik, dan juga sekaligus untuk semua pasangan titik. Ketiadaan garis yang menghubungkan sebuah pasangan dilambangkan dengan tak-hingga. Dalam pengertian lain Algoritma *Floyd-Warshall* adalah Algoritma yang akan memilih satu jalur terpendek dan teraman dari beberapa alternatif jalur yang telah dihasilkan dari proses kalkulasi.

Dalam beberapa kasus, Algoritma *Dijkstra* gagal memberikan solusi terbaik karena kelemahan yang dimilikinya. Di sinilah peran Algoritma *Floyd-Warshall* yang mencoba untuk memberikan solusi optimum yang memiliki pemikiran terhadap konsekuensi yang ditimbulkan dari pengambilan keputusan pada suatu tahap. Algoritma *Floyd-Warshall* mampu mengurangi pencarian keputusan yang tidak mengarah ke solusi. Prinsip yang dipegang oleh algoritma ini adalah prinsip optimalitas, yaitu *jika solusi per-tahap optimal, maka bagian solusi sampai suatu tahap (misalnya tahap ke-i) juga optimal* (Jong, 2006).

2.2.1.1. Deskripsi Algoritma *Floyd-Warshall*

Dasar algoritma *Floyd-Warshall* adalah sebagai berikut:

- a. Asumsikan semua simpul graf berarah G adalah $V = \{1, 2, 3, 4, \dots, n\}$, perhatikan subset $\{1, 2, 3, \dots, k\}$.
- b. Untuk setiap pasangan simpul i, j pada V , perhatikan semua lintasan dari i ke j dimana semua simpul pertengahan diambil dari $\{1, 2, \dots, k\}$, dan p adalah lintasan berbobot minimum diantara semuanya.
- c. Algoritma ini mengeksploitasi relasi antara lintasan p dan lintasan terpendek dari i ke j dengan semua simpul pertengahan berada pada himpunan $\{1, 2, \dots, k-1\}$.
- d. Relasi tersebut bergantung pada apakah k adalah simpul pertengahan pada lintasan p . Berikut adalah *pseudocode* Algoritma *Floyd-Warshall*:

```
function fw (int[1..n,1..n] graph) {  
  // Inisialisasi  
  var int[1..n,1..n] jarak := graph  
  var int[1..n,1..n] sebelum  
  for i from 1 to n  
    for j from 1 to n  
      if jarak[i,j] < Tak-hingga  
        sebelum[i,j] := i  
  // Perulangan utama pada algoritma  
  for k from 1 to n  
    for i from 1 to n  
      for j from 1 to n  
        if jarak[i,j] > jarak[i,k] + jarak[k,j]  
          jarak[i,j] = jarak[i,k] + jarak[k,j]  
          sebelum[i,j] = sebelum[k,j]  
  return jarak  
}
```

Algoritma 2.1. *Pseudocode Algoritma Floyd-Warshall*

2.2.1.2. Karakteristik Algoritma *Floyd-Warshall*

Beberapa karakteristik yang dimiliki oleh algoritma Floyd-Warshall antara lain (Novandi, 2010):

1. Persoalan dibagi atas beberapa tahap.
2. Ketika masuk ke suatu tahap, hasil keputusan akan menjadi simpul baru.
3. Bobot pada suatu tahap akan meningkat secara teratur seiring bertambahnya jumlah tahapan.
4. Bobot yang ada pada suatu tahap tergantung dari bobot tahapan yang telah dilewati dan bobot pada tahap itu sendiri.
5. Keputusan terbaik pada suatu tahap berkaitan terhadap keputusan pada tahap sebelumnya.
6. Terdapat hubungan yang menyatakan bahwa keputusan terbaik dalam setiap status pada tahap k akan memberikan keputusan terbaik untuk setiap status pada tahap $k + 1$.
7. Prinsip optimalitas berlaku pada algoritma ini.

2.2.1.3. Pemecahan Masalah Algoritma *Floyd-Warshall*

Algoritma *Floyd-Warshall* membandingkan semua kemungkinan lintasan pada graf untuk setiap garis dari semua titik. Misalkan terdapat suatu graf G dengan simpul-simpul V yang masing-masing bernomor 1 sampai n (sebanyak n buah). Misalkan pula terdapat suatu fungsi *shortestPath* (i, j, k) yang mengembalikan kemungkinan jalur terpendek dari i ke j dengan hanya memanfaatkan simpul 1 sampai k sebagai titik perantara. Tujuan akhir penggunaan fungsi ini adalah untuk mencari jalur terpendek dari setiap simpul i ke simpul j dengan perantara simpul 1 sampai $k+1$.

Ada dua kemungkinan yang terjadi:

1. Jalur terpendek yang sebenarnya hanya berasal dari simpul-simpul yang berada antara 1 hingga k .
2. Ada sebagian jalur yang berasal dari simpul-simpul i sampai $k+1$, dan juga dari $k+1$ hingga j

Perlu diketahui bahwa jalur terpendek dari i ke j yang hanya melewati simpul 1 sampai k telah didefinisikan pada fungsi *shortestPath*(i, j, k) dan telah

jelas bahwa jika ada solusi dari i sampai $k+1$ hingga j , maka panjang dari solusi tadi adalah jumlah dari jalur terpendek dari i sampai $k+1$ (yang melewati simpul-simpul 1 sampai k), dan jalur terpendek dari $k+1$ sampai j (juga menggunakan simpul-simpul dari 1 sampai k). Maka dari itu, rumus untuk fungsi *shortestPath*(i, j, k) bisa ditulis sebagai suatu notasi sebagai berikut.:

Basis-0

```
shortestPath (i, j, 0) = edgeCost
(i, j);
```

Rekurens

```
shortestPath (i, j, k) = min
(shortestPath (i, j, k-1) , shortestPath
(i, k, k-1) + shortestPath (k, j, k-1));
```

Algoritma 2.2. Pemecahan Solusi Rute Terpendek

Rumus ini adalah inti dari Algoritma *Floyd-Warshall*, algoritma ini bekerja dengan menghitung *shortestPath*($i,j,1$) untuk semua pasangan (i,j), kemudian hasil tersebut akan digunakan untuk menghitung *shortestPath*($i,j,2$) untuk semua pasangan (i,j), dan seterusnya. Proses ini akan terus berlangsung hingga $k = n$ dan kita telah menemukan jalur terpendek untuk semua pasangan (i,j) menggunakan simpul-simpul perantara.

2.4. Global Positioning System (GPS)

Global Positioning System (GPS) adalah sistem satelit navigasi dan penentuan posisi yang dimiliki dan dikelola oleh Amerika Serikat. Sistem ini didesain untuk memberikan posisi seseorang dan banyak orang secara terus-menerus tanpa bergantung waktu dan cuaca. Saat ini GPS sudah banyak digunakan orang di seluruh dunia dalam berbagai bidang aplikasi yang menuntut informasi tentang posisi, kecepatan, percepatan ataupun waktu yang teliti. GPS dapat memberikan informasi posisi dengan ketelitian bervariasi dari beberapa millimeter sampai dengan puluhan meter. Dengan GPS kita dapat mengetahui posisi geografis kita (lintang, bujur, dan ketinggian di atas permukaan laut), jadi

dimanapun kita berada di muka bumi ini, kita dapat mengetahui posisi kita yang tepat (Abi, 2011).



Gambar 2.2. Contoh-Contoh GPS

Contoh penggunaan GPS meliputi:

- a. Mengetahui batas kecepatan kendaraan.
- b. Dapat mengetahui waktu tempuh yang diperlukan untuk sampai ke tempat tujuan.
- c. Dapat mengetahui tempat atau lokasi daerah.
- d. Melacak dan memantau kendaraan.
- e. Mengetahui *history* (masa lampau) yang dilalui oleh kendaraan
- f. Menjadi solusi keamanan bagi kendaraan pribadi maupun perusahaan dari ancaman pencurian.

2.4.1. Keunggulan GPS

Ada beberapa hal yang membuat GPS menarik untuk digunakan dalam penentuan posisi, seperti yang akan diberikan berikut ini. Patut dicatat disini bahwa beberapa faktor yang disebutkan di bawah ini juga akan berlaku untuk aplikasi-aplikasi GPS yang berkaitan dengan penentuan parameter selain posisi seperti kecepatan, percepatan, maupun waktu yang pada dasarnya juga bisa diberikan oleh GPS (Tanoe, 2011)

1. GPS dapat digunakan setiap saat tanpa bergantung waktu dan cuaca, GPS dapat digunakan baik pada siang maupun malam hari, dalam kondisi cuaca yang buruk sekalipun seperti hujan ataupun kabut. Karena karakteristiknya ini maka penggunaan GPS dapat meningkatkan efisiensi dan fleksibilitas dari pelaksanaan aktivitas-aktivitas yang terkait dengan penentuan posisi, yang pada akhirnya dapat diharapkan akan dapat memperpendek waktu pelaksanaan aktivitas.
2. Satelit-satelit GPS mempunyai ketinggian orbit yang cukup tinggi, yaitu sekitar 20.000 km di atas permukaan bumi, dan jumlahnya relatif cukup banyak, yaitu 24 satelit. Ini menyebabkan GPS dapat meliputi wilayah yang cukup luas, sehingga akan dapat digunakan oleh banyak orang pada saat yang sama, serta pemakaiannya menjadi tidak bergantung pada batas-batas politik dan batas alam. Selama yang bersangkutan mempunyai alat penerima sinyal (*receiver*) GPS, maka ia akan dapat menggunakan GPS untuk penentuan posisi dimana si pengguna GPS berada.
3. Penggunaan GPS dalam penentuan posisi tidak memerlukan adanya saling keterlihatan antara satu titik dengan titik lainnya. Yang diperlukan dalam penentuan posisi titik dengan GPS adalah saling keterlihatan antara titik tersebut dengan satelit. Oleh sebab itu topografi antara titik tersebut sama sekali tidak akan berpengaruh, kecuali untuk hal-hal yang sifatnya non-teknis seperti pergerakan personil dan pendistribusian logistik. Karena karakteristiknya ini, penggunaan GPS akan sangat efisien dan efektif untuk diaplikasikan pada survei dan pemetaan di daerah-daerah yang kondisi topografinya.

4. Posisi yang ditentukan dengan GPS akan mengacu ke suatu datum global, yang dinamakan WGS 1984. Atau dengan kata lain posisi yang diberikan oleh GPS akan selalu mengacu ke datum yang sama. Karakteristik ini sangat menguntungkan untuk kondisi Indonesia yang wilayahnya sangat luas dan terdiri dari banyak pulau, dimana proses penghubung kerangka-kerangka titik di satu pulau dengan titik di pulau lainnya akan sangat sulit atau bahkan tidak mungkin dilakukan kalau kita menggunakan metode terestris. Dalam hal ini seandainya GPS digunakan untuk penentuan posisi, maka survei dan pemetaan yang dilakukan di Jawa misalnya, akan memberikan posisi titik-titik yang datumnya sama dengan titik-titik yang diperoleh dari survei dan pemetaan di Irian Jaya, meskipun tidak ada hubungan langsung secara langsung antara kedua survei GPS yang bersangkutan.
5. GPS dapat memberikan ketelitian posisi yang spektrumnya cukup luas. Dari yang sangat teliti (orde militer) sampai yang biasa-biasa saja (orde puluhan meter). Luasnya spektrum ketelitian yang bisa diberikan ini memungkinkan penggunaan GPS secara efektif dan efisien sesuai dengan ketelitian yang diminta serta dana yang tersedia. Disamping itu, dengan spektrum ketelitian yang begitu luas GPS juga akan bermanfaat untuk banyak bidang aplikasi.

2.4.2. *Global Positioning System (GPS) dan Assisted- Global Positioning System (A-GPS)*

Kemajuan teknologi dapat kita rasakan dari berbagai aspek dalam kehidupan sehari-hari. Dari transportasi, komunikasi, produksi, dan berbagai aspek teknologi lainnya. Seperti penentuan arah atau sistem navigasi juga mengalami berbagai perkembangan. Jika dahulu kita dapat mengetahui lokasi dengan menggunakan kompas, kini hal tersebut berubah menjadi alat navigasi yang lebih canggih dan sistem navigasi tersebut adalah GPS.

GPS sendiri memiliki 2 jenis yaitu GPS dan A-GPS, dimana sistem GPS ini sering kita jumpai pada ponsel-ponsel yang ada dipasaran sekarang ini. Dibawah ini akan dijelaskan apa perbedaan antara GPS dan A-GPS.

A. *Global Positioning System (GPS)*

Perangkat GPS biasanya merupakan perangkat khusus untuk GPS, bukan ponsel dan cara menentukan lokasi GPS tersebut adalah dibutuhkan minimal 3 satelit yang membentuk segitiga untuk mendapatkan lokasi yang akurat. Dari proses tersebut akan didapatkan hasil berupa *Time To First Fix (TTFF)* atau berapa lama sebelum posisi kita terdeteksi. Dibutuhkan waktu mulai dari 30 detik hingga 12 menit untuk menentukan lokasi ketika perangkat khusus GPS dinyalakan. Cukup lama namun setidaknya pengguna tidak perlu memiliki akses ke operator ponsel (pulsa) untuk mengoperasikan GPS.

Waktu tersebut tergantung pada lokasi Anda, jumlah gangguan dan kondisi cakrawala (cuaca). Pada daerah terbuka akan lebih cepat mendapatkan sinyal dibandingkan di daerah perkotaan yang banyak gedung-gedung tinggi atau posisi kita yang berada di dalam ruangan yang dapat mengganggu penerimaan sinyal satelit. Data waktu dari 3 satelit tersebut akan dikomputasi oleh modul GPS di ponsel dan akhirnya akan dihasilkan informasi tentang posisi yang berupa latitude & longitude dan lokasi dalam peta.

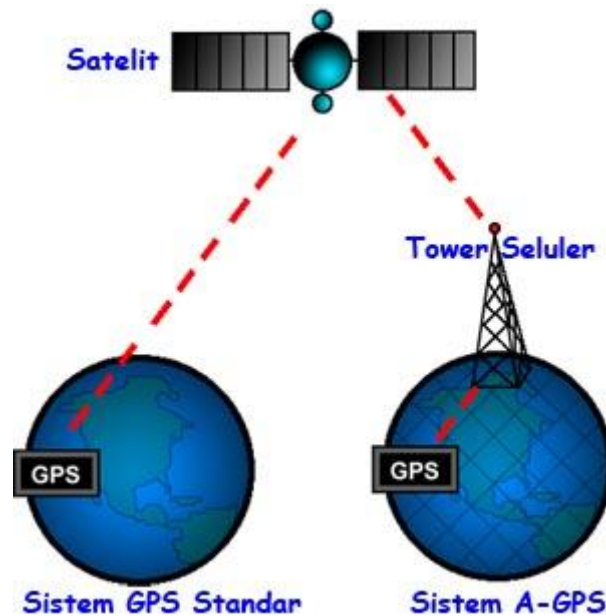
GPS membutuhkan 3 komponen dalam proses penentuan posisi, yaitu Satelit, *Receiver* GPS, dan posisi yang baik (bebas halangan). Perangkat GPS menangkap 3 sinyal dari 24 satelit GPS yang ada di luar angkasa. Supaya perangkat GPS bisa menangkap sinyal dengan baik, perangkat harus berada di luar ruangan, bahkan harus dibawah langit terbuka. Kekuatan sinyal bisa berkurang kalau perangkat GPS berada di bawah pohon, dibawah gedung-gedung pencakar langit, di dalam kendaraan dan sinyal hampir bisa dipastikan menghilang kalau perangkat GPS ada di dalam gedung.

B. *Assisted- Global Positioning System (A-GPS)*

A-GPS merupakan suatu teknologi penyempurnaan dari GPS. Teknologi ini menggunakan *server* bantuan untuk mempercepat waktu penentuan posisi lokasi GPS tersebut, sehingga tidak perlu lagi proses komputasi data dari 3 satelit secara langsung dari ponsel. A-GPS membutuhkan 3 komponen dalam proses penentuan posisi, yaitu Satelit, *Assistance Server (Operator)*, *Receiver A-GPS*.

Karena A-GPS dan *server* bantuan berbagi tugas dalam penentuan posisi, maka proses akan lebih cepat dan lebih efisien dibanding GPS biasa.

Untuk lebih jelasnya anda dapat melihat pada Gambar 2.3 berikut ini:



Gambar 2.3. Penentuan Posisi GPS dan A-GPS

2.5. Android

Android adalah sistem operasi *Mobile Phone* berbasis Linux. Android bersifat *open source* yang *source code*-nya diberikan secara gratis bagi para pengembang untuk menciptakan aplikasi mereka agar dapat berjalan di Android. Pada mulanya, Android adalah salah satu produk besutan dari Android Inc, namun Google mengakuisisi Android Inc, dan semua kekayaan intelektual milik Android Inc. diperoleh Google Inc. yang kemudian mengembangkan kembali sistem Android.

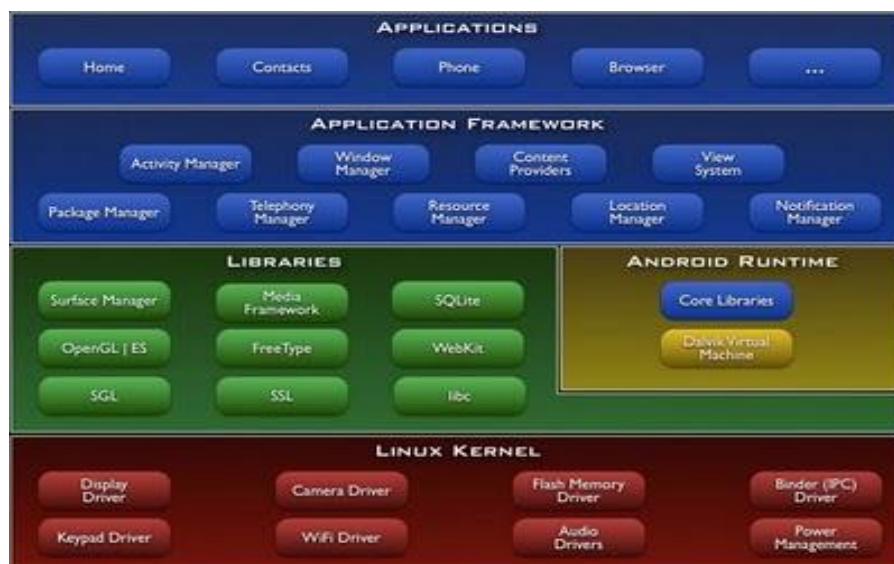
Android Inc. adalah pendatang baru dalam hal membuat software untuk ponsel yang berada di Palo Alto, California, Amerika Serikat. Kemudian dibentuk *Open Handset Alliance*, konsorsium yang terdiri dari 34 perusahaan hardware, software, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, Nvidia, dan lain-lain. *Open Handset Alliance* dibentuk untuk mengembangkan Android yang notabenenya adalah *Operating System*

Open Source pertama untuk Mobile Phone. Pada tanggal 5 November 2007, dirilislah Android versi awal dimana Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan *open source* pada perangkat selular. Dilain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan *open source* perangkat selular.

Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Services* (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution* (OHD). (Safaat, 2011)

2.5.1. Arsitektur Android

Secara garis besar arsitektur Android dapat dilihat pada Gambar 2.4.



Gambar 2.4. Arsitektur Android

A. *Applications* dan *Widgets*

Applications dan *widgets* ini adalah layer dimana kita berhubungan dengan aplikasi saja, dimana biasanya kita *download* aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut. Di layer terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman Java.

B. *Applications Framework*

Android adalah “*Open Development Platform*” yaitu Android menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi resources, menjalankan *service background*, dan sebagainya. Pengembang memiliki akses penuh menuju API framework seperti yang dilakukan oleh aplikasi yang kategori inti. Arsitektur aplikasi dirancang supaya kita dengan mudah dapat menggunakan kembali komponen yang sudah digunakan.

Sehingga bisa kita simpulkan *Applications Framework* ini adalah *layer* dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi Android, karena pada layer inilah aplikasi dapat dirancang dan dibuat, seperti *content-providers* yang berupa sms dan panggilan telepon.

Komponen-komponen yang termasuk di dalam *Applications Frameworks* adalah sebagai berikut :

- a. *Views*
- b. *Content Provider*
- c. *Resource Manager*
- d. *Notification Manager*
- e. *Activity Manager*

C. *Libraries*

Libraries ini adalah layer dimana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Berjalan di atas kernel, layer ini meliputi berbagai *library C/C++* ini seperti *Libc* dan *SSL*, serta :

- a. *libraries* media untuk pemutaran media audio dan video
- b. *libraries* untuk manajemen tampilan *libraries graphics* mencakup *SGL* dan *OpenGL* untuk grafis 2D dan 3D
- c. *libraries SQLite* untuk dukungan database

- d. *libraries* SSL dan *WebKit* terintegrasi dengan *web browser* dan *security*
- e. *libraries LiveWebcore* mencakup modern *web browser* dengan *engine embeded web view*
- f. *libraries* 3D yang mencakup implementasi OpenGL ES 1.0 API's

D. *Android Run Time*

Layer yang membuat aplikasi android dapat dijalankan dimana dalam prosesnya menggunakan implementasi Linux. *Dalvik Virtual Machine* (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi Android. Di dalam Android Run Time dibagi menjadi dua bagian yaitu :

- a. *Core Libraries* : aplikasi Android dibangun dalam bahasa java, sementara Dalvik sebagai virtual mesinnya bukan *Virtual Machine Java*, sehingga diperluas sebuah *libraries* yang berfungsi untuk menerjemahkan bahasa java/C yang ditangani oleh *Core Libraries*.
- b. *Dalvik Virtual Machine* : Virtual mesin berbasis register yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien dimana merupakan pengembangan yang mampu membuat linux kernel untuk melakukan *threading* dan manajemen tingkat rendah.

E. *Linux Kernel*

Linux kernel adalah *layer* dimana inti dari *operating system* dari Android itu berada. Berisi file-file system yang mengatur sistem *processing*, *memory*, *resource*, *drivers*, dan sistem-sistem Android lainnya. Linux kernel yang digunakan Android adalah linux kernel 2.6. (Safaat, 2011)

2.6. **Analisa dan Perancangan Berorientasi Objek**

Teknologi objek menganalogikan sistem aplikasi seperti kehidupan nyata yang didominasi oleh objek. Didalam membangun sistem berorientasi objek akan menjadi lebih baik apabila langkah awalnya didahului dengan proses analisis dan perancangan yang berorientasi objek. Tujuannya adalah untuk mempermudah *programmer* didalam mendesain program dalam bentuk objek-objek dan

hubungan antar objek tersebut untuk kemudian dimodelkan dalam sistem nyata (Widodo, 2011).

Perusahaan *software*, *Rational Software*, telah membentuk konsorsium dengan berbagai organisasi untuk meresmikan pemakaian *Unified Modelling Language* (UML) sebagai bahasa standar dalam *Object Oriented Analysis Design* (OOAD).

2.6.1. *Unified Modelling Language* (UML)

Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem (Widodo, 2011).

Untuk merancang sebuah model, UML memiliki beberapa diagram antara lain : *use case diagram*, *class diagram*, *statechart diagram*, *activity diagram*, *sequence diagram*, *collaboration diagram*, *component diagram*, *deployment diagram*.

2.6.2. *Use Case Diagram*

Use case diagram merupakan sebuah gambaran fungsionalitas sebuah sistem. Sebuah *use case* merepresentasikan interaksi antara aktor dengan sistem. *Use case* sangat menentukan karakteristik sistem yang sedang dibuat. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu (Widodo, 2011).

Dalam sebuah sistem *use case diagram* akan sangat membantu dalam hal menyusun *requirement*, mengkomunikasikan rancangan dengan klien dan merancang *test case* untuk semua fitur yang ada pada sistem.

2.6.3. *Class Diagram*

Class merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi) (Widodo, 2011).

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok yaitu nama, *stereotype*, atribut dan metoda.

2.5.4. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang *trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan (Dharwiyanti, 2006).

2.7. Rational Unified Process (RUP)

Untuk pengembangan aplikasi pencarian rute terpendek lokasi fasilitas umum berbasis Android pada tugas akhir ini menggunakan metode pengembangan perangkat lunak *Rational Unified Process* (RUP).

2.7.1. Pengertian RUP

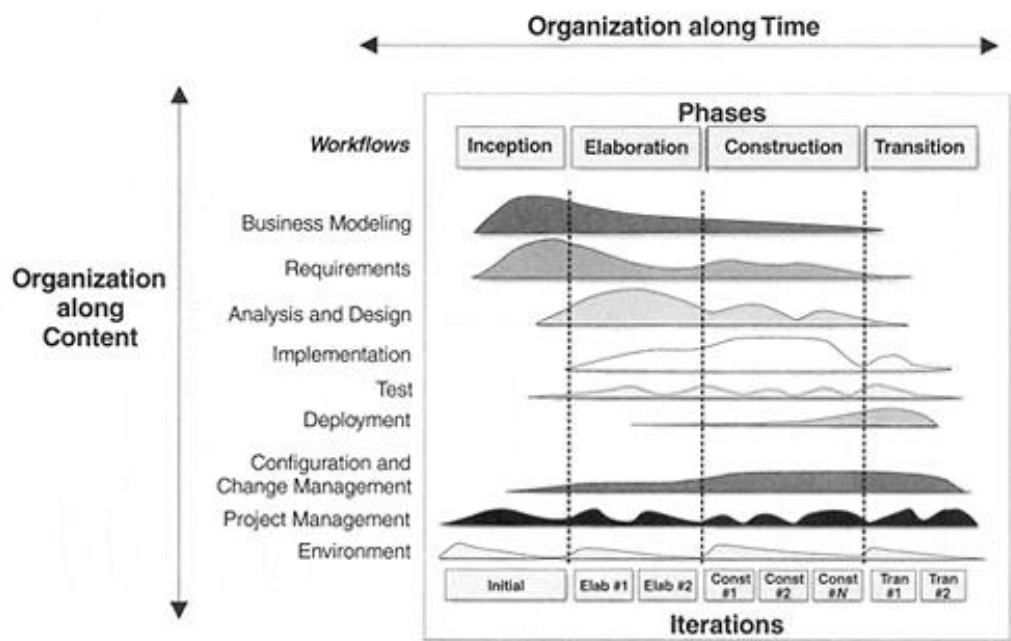
Rational Unified Process adalah sebuah Proses Rekayasa Perangkat Lunak. RUP menyediakan pendekatan disiplin untuk memberikan tugas dan tanggung jawab dalam organisasi pengembang perangkat lunak. Tujuannya untuk memastikan perangkat lunak yang berkualitas tinggi dan sesuai kebutuhan penggunaanya dalam anggaran dan jadwal yang dapat diprediksi (Kruchten, 2000).

RUP mengarahkan kita terhadap pengembangan perangkat lunak secara praktis dan efektif. Terdapat 6 *best practice* atau disebut juga *basic principle* dalam metode RUP, antara lain (Kruchten, 2000):

1. *Develop software iteratively*, bertujuan untuk mengurangi resiko pada awal proyek.

2. *Manage requirements*, bertujuan untuk mengatur kebutuhan yang diperlukan selama proyek.
3. *Use component-based architectures* untuk membangun komponen arsitektur sebuah proyek.
4. *Visually model software*, bertujuan untuk merancang sebuah model visual perangkat lunak, untuk mendapatkan struktur dan perilaku dari aritektur perangkat lunak.
5. *Continuously verify software quality*.
6. *Control changes to software*. kemampuan untuk mengatur serta mengubah perangkat lunak saat dibutuhkan.

RUP menggunakan konsep *object oriented*, dengan aktifitas yang berfokus pada pengembangan model dengan menggunakan *Unified Model Language* (UML). Melalui Gambar 2.4 dibawah dapat dilihat bahwa RUP memiliki 2 dimensi, yaitu:



Gambar 2.5. Struktur Proses 2 Dimensi RUP

Dimensi pertama digambarkan secara horizontal. Dimensi ini mewakili aspek-aspek dinamis dari pengembangan perangkat lunak. Aspek ini dijabarkan dalam tahapan pengembangan atau fase. Setiap fase akan memiliki suatu *major*

milestone yang menandakan akhir dari awal dari phase selanjutnya. Setiap phase dapat berdiri dari satu atau beberapa iterasi. Dimensi ini terdiri atas *Inception*, *Elaboration*, *Construction*, dan *Transition*.

Dimensi kedua digambarkan secara vertikal. Dimensi ini mewakili aspek-aspek statis dari proses pengembangan perangkat lunak yang dikelompokkan ke dalam beberapa disiplin. Proses pengembangan perangkat lunak yang dijelaskan kedalam beberapa disiplin terdiri dari empat elemen penting, yakni *who is doing*, *what*, *how* dan *when*. Dimensi ini terdiri atas *Business Modeling*, *Requirement*, *Analysis and Design*, *Implementation*, *Test*, *Deployment*, *Configuration* dan *Change Manegement*, *Project Management*, *Environtment*.

2.7.2. Fase RUP

Fase-fase pada RUP berdasarkan waktu pengerjaan proyek dapat dibagi menjadi 4 fase, yaitu *Inception*, *Elaboration*, *Construction* dan *Transition* (Rational Team, 2001).

1. Fase *Inception*

Fase *inception* merupakan fase untuk mengidentifikasi masalah, untuk itu diperlukan juga idetifikasi entitas dari luar yang berhubungan dengan sistem. Pada fase ini melibatkan semua identifikasi *use case* dan gambaranya. Selain itu juga termasuk kriteria keberhasilan proyek, perkiraan resiko, perkiraan terhadap *resource* yang dibutuhkan dan merencanakan penjadwalan *milestone*. Hasil yang diperoleh pada fase ini adalah :

- a. Dokumen visi (visi dari kebutuhan proyek, kata kunci, batasan utama).
- b. Inisialisasi model *use-case* (10%-20% selesai).
- c. Daftar kata.
- d. *Business case*, termasuk didalamnya konteks bisnis, kriteria sukses, pengenalan pasar dan proyeksi keuangan.
- e. Inisialisasi penilaian resiko.
- f. Rencana proyek dan menunjukan fase serta iterasi.
- g. Model bisnis jika diperlukan

Kriteria evaluasi untuk fase *Inception* adalah :

- a. Menyesuaikan *stakeholder* dengan *scope definition* dan perkiraan biaya atau perkiraan jadwal.
- b. Pemahaman terhadap *use-case* utama.
- c. Kredibilitas dari perkiraan biaya, jadwal, prioritas, resiko dan proses pengembangan.
- d. Pemahaman terhadap *prototype*.

2. Fase *Elaboration*

Tujuan dari fase *elaboration* (pengembangan) adalah menganalisa area permasalahan, mengembangkan rencana proyek, dan menghilangkan unsur-unsur yang memiliki resiko besar terhadap proyek. Adapun hasil dari fase *elaboration* adalah:

- a. *Use case* model, seluruh use case dan aktor telah teridentifikasi.
- b. *Requirement* tambahan yang mungkin tidak bersifat fungsional bagi proyek.
- c. *Software Architecture Description* (Deskripsi Arsitektur Perangkat Lunak).
- d. Prototipe dari arsitektur yang dapat dieksekusi.
- e. Revisi daftar tingkat resiko dan revisi *business-case*.
- f. Rencana pengembangan keseluruhan proyek.
- g. Persiapan dokumen panduan bagi pengguna (*user manual*).

Kriteria utama dalam fase *elaboration* melibatkan pertanyaan berikut :

- a. Apakah produk sudah stabil ?
- b. Apakah rancangan arsitekturalnya sudah stabil ?
- c. Apakah saat demo prototipe, unsur yang memiliki resiko telah bisa diatur ?
- d. Apakah rencana konstruksi telah detail dan akurat ?
- e. Apakah *stakeholder* bersedia dan menyepakati visi dari pengembangan proyek tersebut?
- f. Apakah pembelanjaan *actual-resource* terhadap rencana pembelanjaan dapat diterima?

3. Fase *Contruction*

Selama fase kontruksi, semua komponen dan fitur yang dikembangkan terintergrasi ke dalam produk dan secara menyeluruh semua fitur telah diuji. Di lain sisi, proses konstruksi adalah sebuah proses *manufacturing*, dimana terdapat penekanan dalam mengelola *resource* dan mengatur operasi untuk mengoptimalkan jadwal dan kualitas. Pada tahap ini pola pikir (*mindset*) mengalami perubahan dari pengembangan *intellectual property* pada fase *Inception* dan *Elaboration*, menjadi pengembangan *deployable product*. Kriteria evaluasi terhadap fase *Construction* ini adalah :

- a. Apakah peluncuran produk cukup baik dan dapat diterima di komunitas pengguna?
- b. Apakah semua *stakeholder* siap untuk beralih ke komunitas pengguna?
- c. Apakah pembelanjaan *actual-resource* terhadap rencana pembelanjaan masih tetap diterima?

4. Fase *Transition*

Tujuan dari fase ini adalah untuk transisi dari produk perangkat lunak ke pengguna akhir. Apabila produk telah di luncurkan kepada pengguna, maka isu-isu akan muncul dari pengguna. Nantinya isu ini akan digunakan untuk tahap perbaikan terhadap produk. Kriteria evaluasi untuk fase *Transition* adalah :

- a. Apakah pengguna merasa puas?
- b. Apakah pembelanjaan *actual-resource* terhadap rencana pembelanjaan masih tetap diterima?

2.8. Sistem Informasi Geografis (SIG)

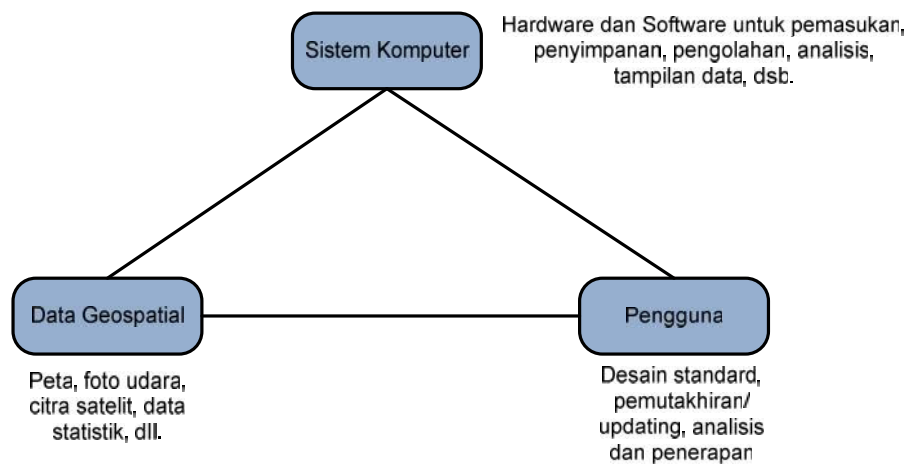
2.8.1. Pengertian Sistem Informasi Geografis (SIG)

Sistem Informasi Geografis (SIG) adalah sistem informasi yang digunakan untuk memasukkan, menyimpan, memanggil, mengolah, menganalisis dan menghasilkan data bereferensi geografis (permukaan bumi) atau data geospasial, untuk mendukung pengambilan keputusan dalam perencanaan dan pengelolaan

penggunaan lahan, sumber daya alam, lingkungan transportasi, fasilitas kota, dan pelayanan umum lainnya.

Sumber lainnya menyatakan Sistem Informasi Geografis itu adalah kumpulan yang terorganisasi dari perangkat keras komputer, perangkat lunak, data geografi dan personil yang dirancang secara efisien untuk memperoleh, menyimpan, meng-*update*, memanipulasi, menganalisis dan menampilkan semua bentuk informasi yang bereferensi geografis (Wiley, 1990).

Komponen dari SIG adalah sistem komputer yang terdiri atas perangkat keras (*hardware*) dan perangkat lunak (*software*), data geospasial dan pengguna (*brainware*) seperti yang ditunjukkan pada Gambar 2.5 (Rizal, 2011).



Gambar 2.6. Komponen Sistem Informasi Geografis

Data yang diolah pada SIG adalah data geospasial (data spasial dan data non-spasial). Biasanya data non-spasial tidak digambarkan karena memang dalam SIG yang dipentingkan adalah tampilan data secara spasial.

Data spasial sendiri itu adalah data yang berhubungan dengan kondisi geografi misalnya sungai, wilayah administrasi, gedung, jalan raya, dan sebagainya. Biasanya data spasial bisa didapatkan dari peta, foto udara, citra satelit, data statistik dan lain-lain. Secara umum persepsi manusia mengenai bentuk representasi entitas spasial adalah konsep raster dan vektor. Sedangkan data non-spasial adalah selain data spasial yaitu data yang berupa text atau angka yang biasa disebut dengan atribut.

Data non-spasial ini akan menerangkan data spasial atau sebagai dasar untuk menggambarkan data spasial. Dari data non-spasial ini nantinya dapat dibentuk data spasial. Misalnya jika ingin menggambarkan peta penyebaran penduduk maka diperlukan data jumlah penduduk dari masing-masing daerah (data non-spasial), dari data tersebut nantinya akan dapat digambarkan pola penyebaran penduduk untuk masing-masing daerah (spasial).

2.8.2. Subsistem SIG

Sistem Informasi Geografis (SIG) dapat diuraikan menjadi beberapa subsistem sebagai berikut (Nirwan, 2011):

1. Data Input

Subsistem ini bertugas untuk mengumpulkan dan mempersiapkan data spasial dan atribut dari berbagai sumber.

2. Data Output

Subsistem ini menampilkan atau menghasilkan keluaran seluruh atau sebagian basis data dalam bentuk *softcopy* maupun bentuk *hardcopy*.

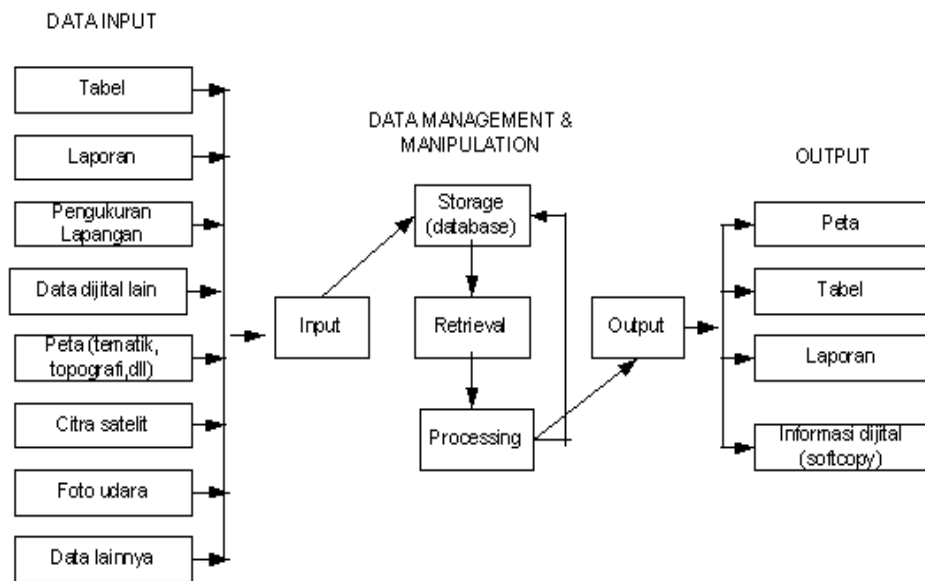
3. Data Management

Subsistem ini mengorganisasi baik data spasial maupun atribut kedalam sebuah basis data sedemikian rupa sehingga mudah dipanggil, di-*update*, dan di-*edit*.

4. Data Manipulation & Analysis

Subsistem ini menentukan informasi-informasi yang dapat dihasilkan oleh SIG. Subsistem ini juga melakukan manipulasi dan pemodelan data untuk menghasilkan informasi yang diharapkan.

Jika subsistem SIG tersebut diperjelas berdasarkan uraian jenis masukan, proses, dan jenis keluaran yang ada didalamnya, maka subsistem SIG dapat juga digambarkan seperti pada Gambar 2.6.



Gambar 2.7. Subsistem Sistem Informasi Geografis

2.8.3. Konsep Model Data Spasial SIG

Data spasial merupakan data yang paling penting dalam SIG. Data spasial ada 2 macam yaitu data raster dan data vektor (Prahasta, 2009):

a. Data Raster

Model data raster menampilkan, menempatkan dan menyimpan data spasial dengan menggunakan struktur matriks atau pixel-pixel yang membentuk grid. Konsep model data ini adalah dengan memberikan nilai yang berbeda untuk tiap-tiap pixel atau grid dari kondisi yang berbeda.

b. Data Vektor

Model data vektor yang menampilkan, menempatkan dan menyimpan data spasial seperti titik-titik, garis-garis, atau kurva atau poligon beserta atribut-atributnya. Bentuk dasar representasi data spasial didalam sistem model data vektor, didefinisikan oleh sistem koordinat kartesian dua dimensi (x,y).

2.8.4. *Universal Tranverse Mercator (UTM)*

UTM merupakan satuan koordinat berdasarkan satuan jarak dan berhubungan dengan proyeksi yang digunakan, yaitu konversi UTM. Proyeksi UTM adalah sistem proyeksi orthometrik dengan satuan panjang meter (m) berdasar *mercator* (bidang silinder) terhadap kedudukan bidang proyeksi *transversal* (melintang), menggunakan zona dengan interval 6° meridian yang dikenalkan oleh Mercator.

Koordinat UTM adalah koordinat ortometrik 2 dimensi, dengan titik acuan absis x dalam satuan E (*East*) awal 500.000 m dan ordinat y dalam satuan N (*North*) awal 10.000.000 m terletak di pusat proyeksi (perpotongan *Meridian Central* (MC) atau tengah zona dengan ekuator). Arah utara *grid* sejajar proyeksi zona MC, merupakan juring elipsoid dengan batasan 6° diawali di Bujur 180° dengan arah Timur (zona 1) sampai dengan zona 60. Artinya berawal di Bujur 190° ketimur (Bujur Timur) melalui Bujur 0° di Greenwich (zona 30) berakhir di Bujur 180 Timur (zona 60) garis Bujur atau garis Meridian. Indonesia terletak pada zona 46 hingga zona 54. Kota Pekanbaru terletak pada zona 47 N (Prahasta, 2009)

Proyeksi potongan satu bidang dengan elipsoid melalui dua kutubnya yang merupakan garis di permukaan elipsoid bumi membujur dari Kutub Utara ke Kutub Selatan, dihitung dari Bujur 0° Greenwich 180° kearah Timur dan 180° kearah Barat.

2.8. Rumus Perhitungan Waktu Tempuh

Dibawah ini adalah rumus untuk melakukan perhitungan kecepatan, jarak dan waktu tempuh (Halliday, 2012):

Rumus:

$$v = \frac{s}{t}$$

Dengan ketentuan:

1. s = Jarak yang ditempuh (m, km)
2. v = Kecepatan (km/jam, m/s)
3. t = Waktu tempuh (jam, detik)

Catatan:

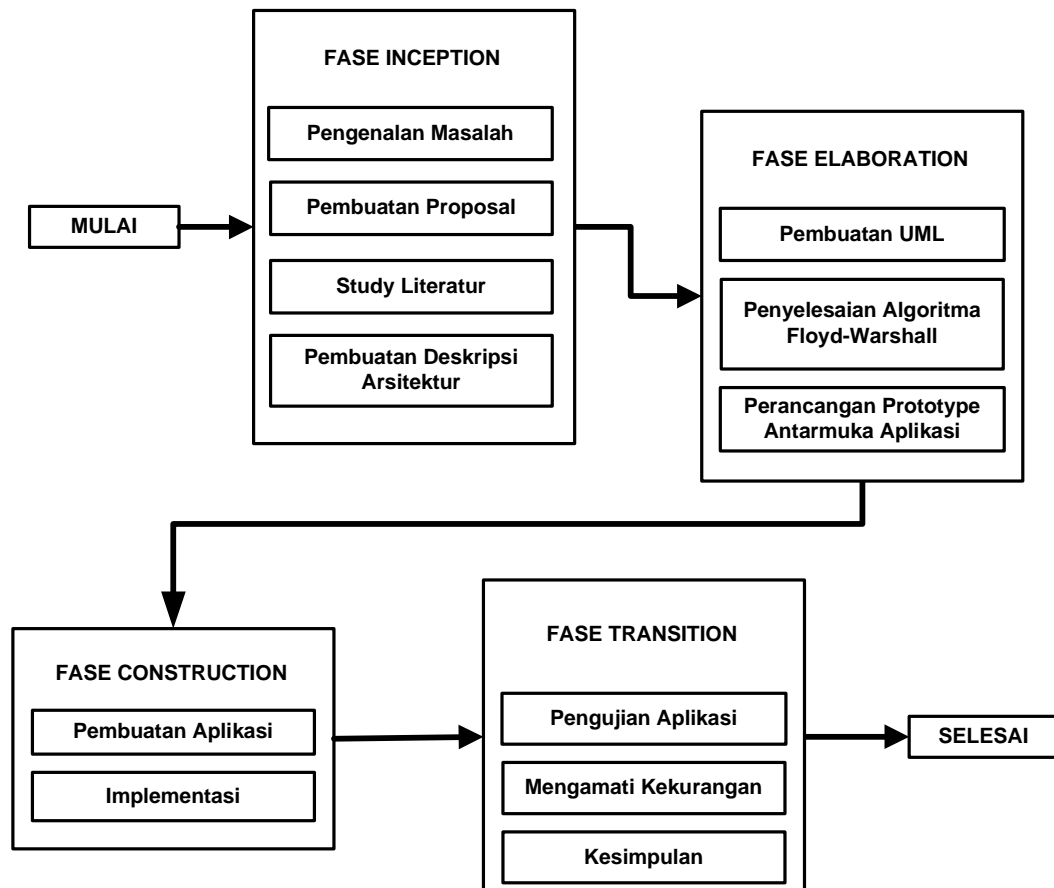
1. Untuk mencari jarak yang ditempuh, rumusnya adalah $s = v \times t$.
2. Untuk mencari waktu tempuh, rumusnya adalah $t = \frac{s}{v}$.
3. Untuk mencari kecepatan, rumusnya adalah $v = \frac{s}{t}$.

BAB III

METODOLOGI PENELITIAN

Pada bab ini akan dipaparkan tentang langkah-langkah yang digunakan untuk membahas permasalahan yang diambil dalam penelitian. Tahapan penelitian yang akan dilaksanakan pada pembuatan aplikasi Pencarian Rute Terpendek Lokasi Fasilitas Umum menggunakan Algoritma *Floyd-Warshall* pada Android dengan menggunakan RUP (*Rational Unified Process*), adalah:

3.1. Alur Tahapan RUP



Gambar 3.1. Tahapan Penelitian

Alur tahapan RUP yang akan digunakan dalam membuat aplikasi pencarian rute terpendek lokasi fasilitas umum menggunakan algoritma *floyd-warshall* pada Android ini dapat dilihat pada Gambar 3.1 dengan penjelasan tiap fase sebagai berikut:

3.1.1. Fase *Inception*

Pada fase ini akan dilakukan tugas-tugas sebagai berikut:

- a. Pengenalan masalah, memahami permasalahan yang terjadi, mengapa diperlukan suatu aplikasi pencarian rute terpendek lokasi fasilitas umum menggunakan algoritma *floyd-warshall* di Kota Pekanbaru pada perangkat *smartphone* yang bersistem operasi android.
 - i. Pengaruh teknologi yang berkembang pesat membuat manusia membutuhkan sarana informasi yang lebih maju dengan memanfaatkan teknologi yang ada.
 - ii. Dibutuhkan waktu yang lebih efektif dan efisien saat melakukan pencarian lokasi fasilitas umum.
 - iii. Belum adanya penelitian yang membahas pencarian rute terpendek dengan menggunakan algoritma *floyd-warshall*.
 - iv. Dibutuhkan penelitian untuk mengetahui konsep dasar, struktur dan sistem kerja, serta pengkodean dalam merancang dan membangun aplikasi pencarian rute terpendek lokasi fasilitas umum pada sistem operasi Android
- b. Pembuatan proposal, yaitu mencakup latar belakang permasalahan, rumusan masalah, batasan penelitian, tujuan, manfaat, sistematika penulisa, landasan teori, dan metodologi penelitian.
- c. Studi Literatur, mencakup penelusuran teori-teori yang berhubungan dengan permasalahan, yang bersumber dari buku, jurnal, artikel internet dan penelitian-penelitian sejenis yang dapat mendukung pemecahan masalah dalam penelitian yang dilakukan.

3.1.2. Fase *Elaboration*

Pada fase *elaboration* akan dilakukan tugas-tugas sebagai berikut :

- a. Pembuatan UML yang meliputi *usecase diagram*, *class diagram*, *activity diagram*, *sequence diagram* dan *deployment diagram*.
- b. Pembuatan *source code* algoritma *floyd-warshall* dan perhitungan manual dari algoritma tersebut.
- c. Pembuatan Prototype Antarmuka Aplikasi.

3.1.3. Fase *Construction*

Fase ini meliputi kegiatan pengkodean, implementasi, dan pengujian. Fase ini dilakukan setelah fase *elaboration* selesai dilakukan, karena fase *construction* bisa dilaksanakan setelah fase sebelumnya selesai dilakukan. Fase ini adalah fase dimana pembuat aplikasi mulai membangun aplikasi berdasarkan hasil fase *inception* dan fase *elaboration*

3.1.4. Fase *Transition*

Setelah menyelesaikan fase *construction*, kemudian langkah selanjutnya yaitu fase *transition*. Fase ini merupakan fase dimana akan dilakukan *deploying* aplikasi untuk melihat kekurangan aplikasi oleh pembuat aplikasi, dan testing tiap fungsi pada aplikasi, kemudian diambil kesimpulan tentang penggunaan aplikasi.

BAB IV

ANALISA DAN PERANCANGAN

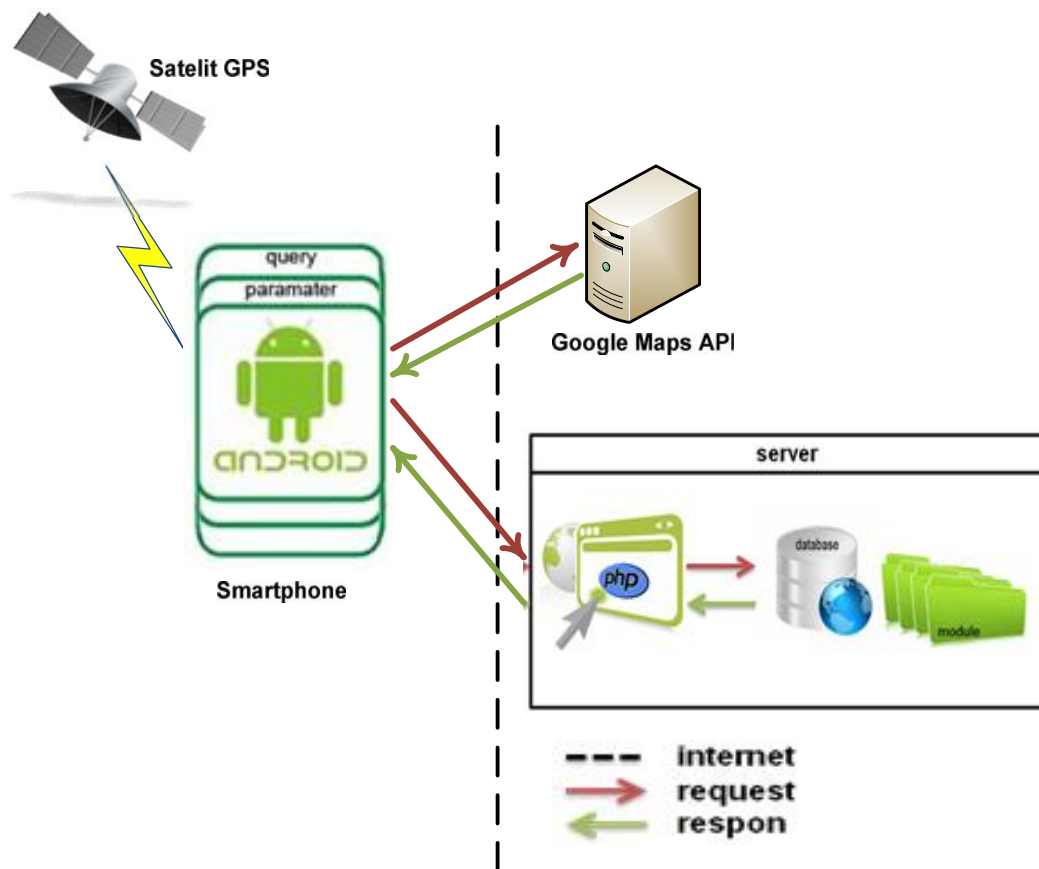
Bab ini merupakan bagian dari fase *inception* dan fase *elaboration*, dimana akan dilakukan analisa sejalan dengan pembuatan deskripsi arsitektur yang dibutuhkan aplikasi sebagai bagian dari fase *inception*, dan kemudian dilakukan pembuatan UML (*Unified Modelling Language*), struktur menu aplikasi yang akan dibangun, dan perancangan *prototype* antarmuka aplikasi yang akan dibangun sebagai bagian dari fase *elaboration*.

4.1. Deskripsi Umum Sistem

Aplikasi pencarian rute terpendek menggunakan Algoritma *Floyd-Warshall* yang dibahas dalam penelitian ini adalah aplikasi yang berbasis *client-server*, dan akan dijalankan pada perangkat *smartphone* dengan sistem operasi Android. Aplikasi ini menggunakan bahasa pemrograman PHP sebagai penghubung dalam pengiriman *request* dan penerimaan respon terhadap *server* yang menggunakan *database MySQL*.

Aplikasi *client* yang akan dibangun merupakan aplikasi yang mampu merequest akses untuk menampilkan lokasi fasilitas umum dari *server* yang meliputi *database* melalui bahasa pemrograman berbasis *web*, kemudian *server* akan memberikan respon kepada *client* melalui jalur pengiriman *request* sebelumnya.

Gambaran umum pencarian rute terpendek menggunakan Algoritma *Floyd-Warshall* ini bertujuan memberikan gambaran mengenai struktur menu dan konsep dasar aplikasi. Untuk lebih jelasnya deskripsi arsitektur sistem ini dapat dilihat pada gambar 4.1. dibawah ini.



Gambar 4.1. Gambaran Umum Sistem

Dari Gambar 4.1 diatas dapat dilihat proses kerja aplikasi yang akan dibuat, ada tiga bagian penting yang saling terhubung dalam kerja sistemnya, diantaranya:

1. *Mobile Device* (Android) merupakan perangkat tempat berjalannya aplikasi pencarian lokasi fasilitas umum. Dari perangkat inilah pengguna berinteraksi dengan sistem dengan memanfaatkan jaringan internet *mobile* untuk mengakses informasi yang diinginkan oleh pengguna.
2. *Database Server*, terdiri dari dua bagian, yaitu:
 - a. Penghubung (PHP), berfungsi sebagai jembatan penghubung antara sistem yang berjalan pada perangkat Android (*client*) dan *database*. Peran penghubung sangat penting, karena sisi *client* tidak bisa langsung menyentuh *database* tanpa perantara. Penghubung ini yang bertugas mengirimkan *request* dan respon antara *client* dan *server*.

- b. *Database*, merupakan bagian yang berfungsi sebagai *database* dari aplikasi pencarian lokasi fasilitas umum. *Database* ini yang bertanggung jawab memberikan respon sesuai *request* dari *client*. *Database* yang digunakan adalah MySQL.

User langsung dapat mendapatkan hak akses aplikasi jika aplikasi ini telah dipasangkan pada perangkat Android. Aplikasi ini tidak membatasi hak akses.

4.2. Algoritma *Floyd Warshall*

4.2.1. Cara Kerja Algoritma *Floyd Warshall*

Algoritma ini mencari panjang lintasan terpendek dari node asal ke node tujuan dalam sebuah graf. Langkah-langkah dalam menentukan lintasan terpendek pada algoritma *floyd warshall* yaitu :

1. Pertama persoalan dibagi atas beberapa tahap dan buat *flowchart* untuk lebih mempermudah pencarian.
2. Ketika masuk ke suatu tahap, hasil pada tahap tersebut akan menjadi simpul baru untuk tahap selanjutnya.
3. Tentukan 1 titik sebagai titik awal agar pencarian algoritma dapat dilakukan.
4. Cari node yang bertetangga langsung dengan titik simpul/titik awal.
5. Bandingkan rute tiap tahap yang bobotnya sudah dijumlahkan dengan bobot-bobot pada tahap sebelumnya, jika sudah maka cari rute dengan bobot yang terkecil sampai proses pencarian berakhir.
6. Bobot yang dimiliki oleh suatu tahap akan dijumlahkan dengan bobot yang ada pada tahap-tahap sebelumnya seiring dengan bertambahnya jumlah tahapan.
7. Pencarian berhenti apabila node tujuan telah ditemukan.
8. Setelah proses selesai, lihat ada berapa rute yang diperoleh untuk ke suatu tujuan tertentu dan pilih rute yang paling kecil untuk menjadi rute terpendek dari algoritma *floyd-warshall*

4.2.2. Perhitungan Manual Algoritma *Floyd Warshall*

Misalkan terdapat suatu graf berbobot yang merepresentasikan kondisi keterhubungan antarkota di suatu daerah, dalam kasus ini kita misalkan seseorang akan melakukan perjalanan dari UIN Suska Riau (Titik A) ke Terminal AKAP (Titik U).



Gambar 4.2. Contoh *Node* Jalan Berbobot

A. *Flowchart Algoritma Floyd-Warshall*

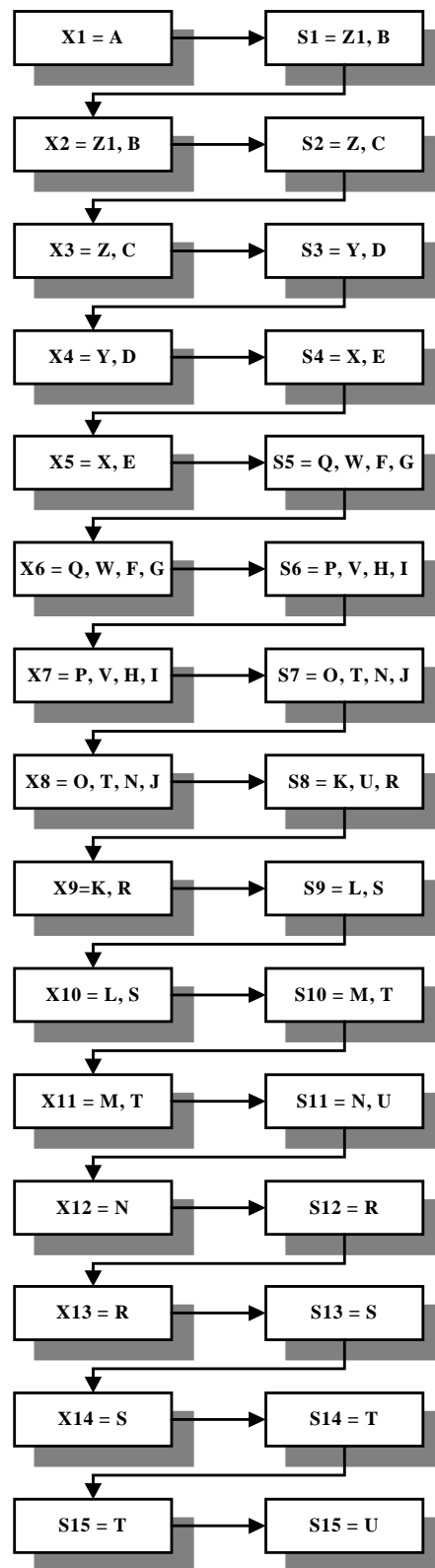
Langkah pertama adalah mengelompokkan proses pencarian setiap tahap dan mencari node yang terhubung langsung dengan titik simpul yang sedang ditinjau, berikut adalah prosesnya:

- Tahap 1: Pada tahap ini titik simpul yang sedang di tinjau adalah A. Kemudian titik A itu sendiri memiliki 2 kandidat solusi yaitu Z1 dan B.
- Tahap 2: Setelah tahap 1 selesai ditinjau, maka sekarang proses yang dilakukan ada pada tahap 2, dimana titik kandidat solusi yang ada pada tahap 1 yaitu Z1 dan B dijadikan titik simpul pada tahap 2, jadi titik simpul pada tahap 2 ini adalah Z1 dan B. Titik Z1 dan B ini memiliki kandidat solusi Z dan C.
- Tahap 3: Setelah tahap 2 selesai ditinjau, maka sekarang proses yang dilakukan ada pada tahap 3, dimana titik kandidat solusi yang ada pada tahap 2 yaitu Z dan C dijadikan titik simpul pada tahap 3. Sehingga titik simpul tahap 3 adalah Z dan C yang memiliki kandidat solusinya adalah Y dan D.
- Tahap 4: Setelah tahap 3 selesai ditinjau, maka sekarang proses yang dilakukan ada pada tahap 4, dimana titik kandidat solusi yang ada pada tahap 3 yaitu Y dan D dijadikan titik simpul pada tahap 4. Sehingga titik simpul tahap 4 adalah Y dan D yang memiliki kandidat solusinya adalah X dan E.
- Tahap 5: Setelah tahap 4 selesai ditinjau, maka sekarang proses yang dilakukan ada pada tahap 5, dimana titik kandidat solusi yang ada pada tahap 4 yaitu X dan E dijadikan titik simpul pada tahap 5. Sehingga titik simpul tahap 5 adalah X dan E yang memiliki kandidat solusinya adalah Q, W, F dan G.
- Tahap 6: Untuk tahap 6 sampai dengan tahap akhir, proses yang dilakukan adalah sama, untuk lebih mengetahui lebih jelas proses yang dilakukan dibawah ini adalah *flowchart* dari seluruh proses pencarian Algoritma *Floyd-Warshall*.

Keterangan gambar:

X_i : Titik Simpul

S_i : Kandidat Solusi



Gambar 4.3. Flowchart Perhitungan Algoritma Floyd-Warshall

B. Analisa Algoritma *Floyd-Warshall*

Dimana:

f : nilai jarak antar titik per-tahap

k : tahap ke- n

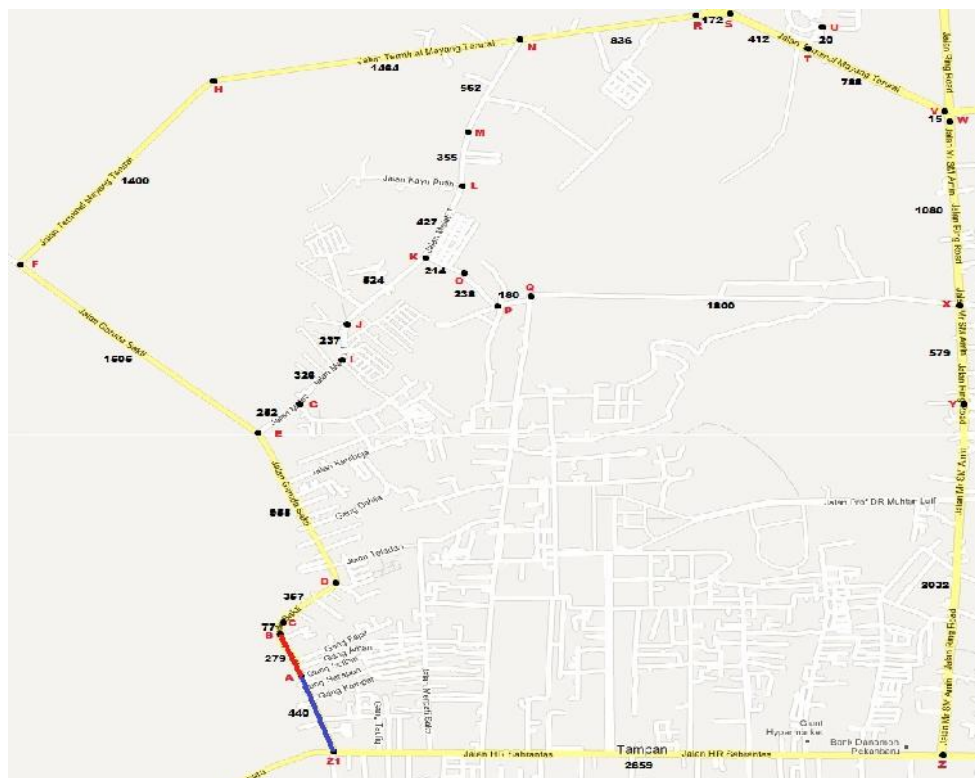
s : titik simpul yang sedang ditinjau

Tahap 1:

Titik Tujuan	Titik Asal	
s1	Solusi Optimum	
	$f1(s)$	$x1$
Z1	440	A
B	279	A

Penjelasan:

Pada tahap pertama ini, algoritma melakukan proses dari titik A ke titik-titik yang saling berhubungan, dimana pada tahap ini titik **A** terhubung dengan titik **Z1** dan titik **B** yang masing-masing mempunyai bobot **440 m** dan **279 m**. Berikut tampilan pencarian rute terpendek pada tahap 1:



Gambar 4.4. Tahap 1

Tahap 2:

Titik Tujuan	Titik Asal			
s2	$f1(s)$		Solusi Optimum	
	Z1	B	$f2(s)$	$x2$
Z	2859	-	2859	Z
C	-	77	77	C

Penjelasan:

Pada tahap 2 ini, titik tujuan yang ada pada tahap 1, berubah menjadi titik simpul pada tahap ini. Sehingga untuk mendapatkan jumlah jarak pada tahap ini maka:

1. Jarak pada tahap 1 yaitu $A \Rightarrow Z1 = 440 \text{ m}$;
2. Jarak pada tahap 2 ini yaitu $Z1 \Rightarrow Z = 2859 \text{ m}$;
3. Maka jumlah jarak yang didapatkan pada tahap ini adalah:

$$(A \Rightarrow Z1) + (Z1 \Rightarrow Z) = 440 \text{ m} + 2859 \text{ m} = 3299 \text{ m}.$$

Kemudian untuk jumlah jarak pada titik yang kedua, yaitu:

1. Jarak pada tahap 1 yaitu $A \Rightarrow B = 279 \text{ m}$;
2. Jarak pada tahap 2 ini yaitu $B \Rightarrow C = 77 \text{ m}$.
3. Maka jumlah jarak yang didapatkan pada tahap ini adalah:

$$(A \Rightarrow B) + (B \Rightarrow C) = 279 \text{ m} + 77 \text{ m} = 356 \text{ m}.$$

Berikut tampilan pencarian rute terpendek pada tahap 2, untuk proses selanjutnya dapat dilihat pada lampiran E:



Gambar 4.5. Tahap 2

4.2.3. Perhitungan Waktu Tempuh

Berikut adalah rumus untuk melakukan perhitungan berapa waktu yang dapat di tempuh pengguna jika menggunakan kecepatan rata-rata 40 km/h.

$$Waktu Tempuh = \left(\frac{jumlah\ jarak / 1000}{40} \right) \times 60$$

Dimana:

- Karena satuan kecepatannya adalah kilometer maka jumlah jarak harus kita rubah dari meter ke kilometer sehingga dibagi 1000.
- 40 merupakan kecepatan rata-rata pengguna.
- Waktu tempuh yang akan dihasilkan adalah dalam satuan menit, karena 1 jam = 60 menit maka hasil pembagiannya akan dikalikan dengan 60.

4.3. Fungsi Sistem

Secara umum fungsi sistem ada dua bagian yaitu sistem yang akan dibangun dari sisi perangkat Android dan media penghubung.

4.3.1. Fungsi Sistem dari Sisi Perangkat Android

Sistem yang akan dibangun dari sisi perangkat Android memiliki fungsi-fungsi sebagai berikut:

1. Menampilkan lokasi fasilitas umum dengan rute terpendek.
2. Menampilkan pencarian lokasi berdasarkan kategori.
3. Menampilkan *About*.
4. Menampilkan *My Location*, lokasi pengguna (*realtime*) yang di tampilkan secara bersamaa dengan lokasi-lokasi fasilitas umum yang ada.

4.3.2. Fungsi Media Penghubung

Media penghubung yang akan dibangun memiliki fungsi untuk mengelola koordinat lokasi fasilitas umum dan koordinat jalan yang telah disimpan pada *database*.

4.4. Deskripsi Pengguna

Pengguna dari sistem ini adalah pengguna yang dalam hal ini sebagian besar adalah masyarakat, pengguna diberikan akses penuh terhadap semua menu dan fungsi yang ada pada aplikasi ini. Untuk lebih jelasnya dapat di lihat di tabel 4.1. di bawah ini.

Tabel 4.1. Deskripsi Pengguna.

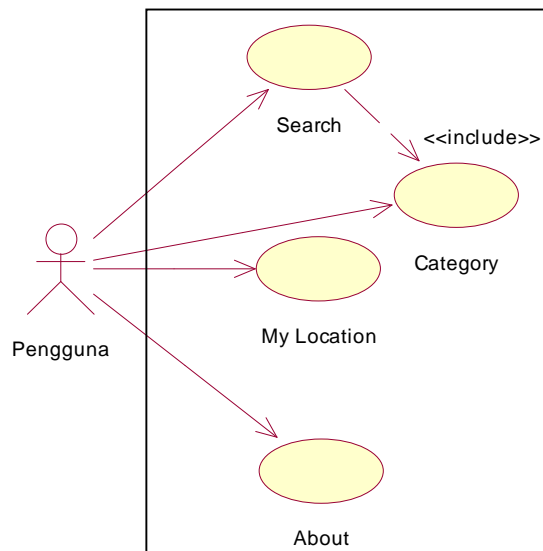
No	Kategori Pengguna	Hak Akses	Keterangan
1	Pengguna (Perangkat Android)	a. Mendapatkan info lokasi <i>ter-update</i> b. Melakukan pencarian lokasi dengan rute terpendek c. Melihat menu <i>about</i> d. Melihat menu <i>my location</i>	Hak akses penuh

4.5. Perancangan Sistem

Setelah dilakukan beberapa tahapan dalam analisa sistem, maka dapat dilakukan beberapa perancangan aplikasi pencarian rute terpendek berbasis *client-server* pada sistem operasi Android. Perancangan-perancangan yang akan dijelaskan dalam laporan ini meliputi perancangan model dalam bentuk UML (*Unified Modeling Language*) yang terdiri dari *Usecase Diagram*, *Class Diagram*, *Activity Diagram*, dan *Sequence Diagram*. Selain itu juga ada perancangan *interface* sistem yang terdiri dari perancangan *prototype* dan struktur menu.

4.5.1. Model Use Case

Usecase diagram merupakan suatu aktivitas yang menggambarkan urutan interaksi antar satu atau lebih aktor dan sistem. *Usecase* yang akan dirancang yaitu *usecase diagram* untuk pengaksesan melalui perangkat Android. Gambar 4.2 dibawah ini menjelaskan aliran *usecase diagram* pengaksesan melalui perangkat Android.



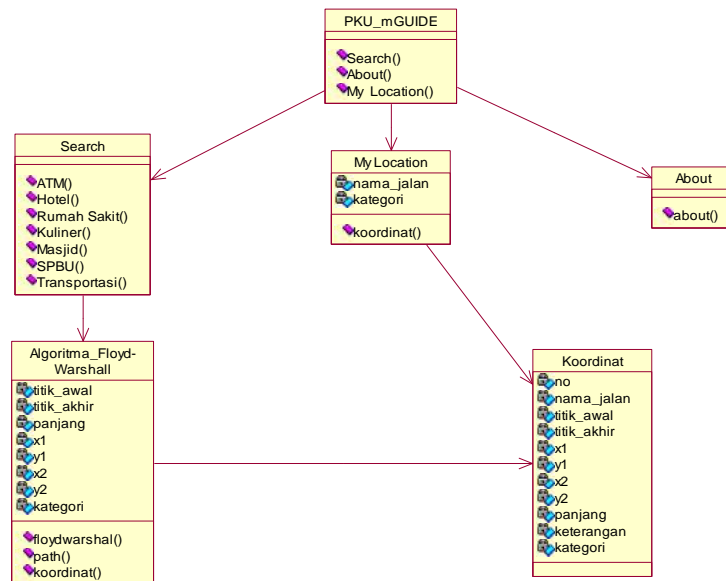
Gambar 4.6. *Use Case Diagram*

Tabel 4.2. Spesifikasi *Use Case Diagram*

No.	Aktor	Nama <i>Usecase</i>	Deskripsi
1.	Pengguna	Search	- Pengguna memilih menu search untuk mencari lokasi fasilitas umum terdekat
		Category	Pengguna memilih kategori fasilitas umum yang ingin dicari agar lebih mudah dan efisien
		My Location	Pengguna melihat lokasi fasilitas umum yang ada di sekitarnya
		About	Pengguna mendapat keterangan mengenai aplikasi.

4.5.2. *Class Diagram*

Class Diagram menggambarkan struktur dan deskripsi *class*, *package*, dan objek yang saling terhubung. *Class diagram* yang dijelaskan pada analisa ini adalah *class diagram* sistem yang terpasang pada perangkat Android. Gambar 4.4 dibawah ini menjelaskan *class diagram* sistem yang terpasang pada perangkat Android.



Gambar 4.7. *Class Diagram* PKU mGUIDE

Berikut adalah deskripsi dan penjelasan dari kelas-kelas yang muncul pada diagram perancangan kelas :

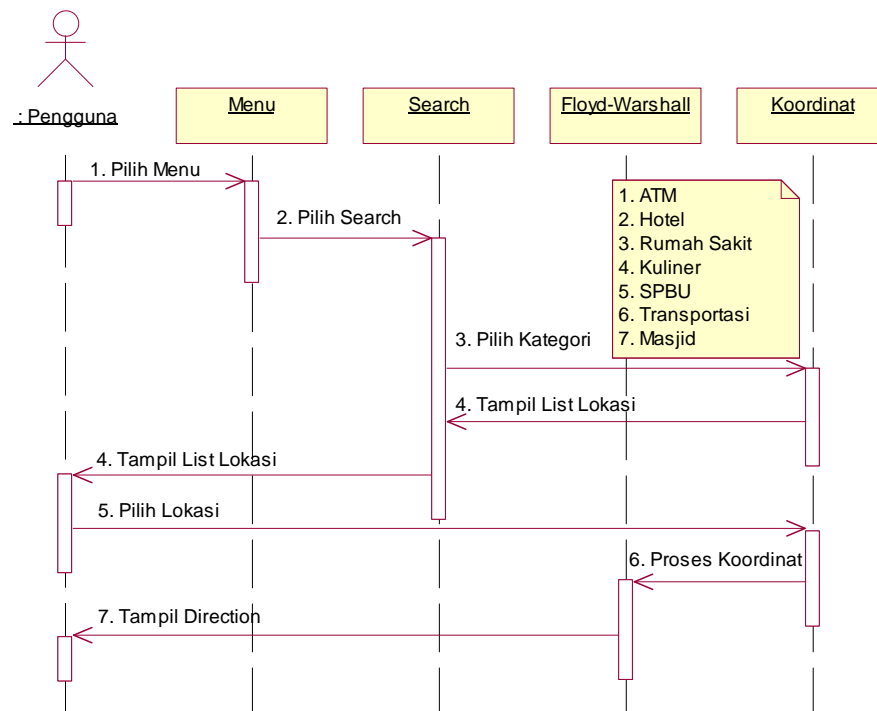
Tabel 4.3. Deskripsi Perancangan *Class Diagram*

Nama Kelas	Nama File	Keterangan
PKU_mGUIDE	PKUmGuide.java	Kelas ini merupakan tampilan utama dari aplikasi PKU mGUIDE.
Search	search.java	Kelas ini merupakan tampilan untuk melakukan pencarian lokasi dengan rute terpendek.
My Location	myLocation.java	Kelas ini merupakan tampilan untuk melihat lokasi fasum yang ada.
Koordinat	fasum.sql	Koordinat merupakan <i>entity</i> dari database fasilitas umum.
About	about.java	Kelas ini untuk menampilkan informasi tentang dari aplikasi
Algoritma_Floyd Warshall	floydw.php	Kelas ini merupakan kelas algoritma

4.5.3. Sequence Diagram

Sequence diagram adalah representasi dari interaksi-interaksi objek yang berjalan pada sistem. Dengan menggunakan *sequence diagram* kita dapat melihat bagaimana objek-objek bekerja. *Sequence diagram* dapat menampilkan bagaimana sistem merespon setiap kejadian atau permintaan dari user, dapat mempertahankan integritas internal, bagaimana data dipindah ke *user interface* dan bagaimana objek-objek diciptakan dan dimanipulasi. Untuk *Sequence Diagram* lainnya dapat di lihat pada lampiran A.

A. Sequence Diagram Search



Gambar 4.8. *Sequence Diagram Search*

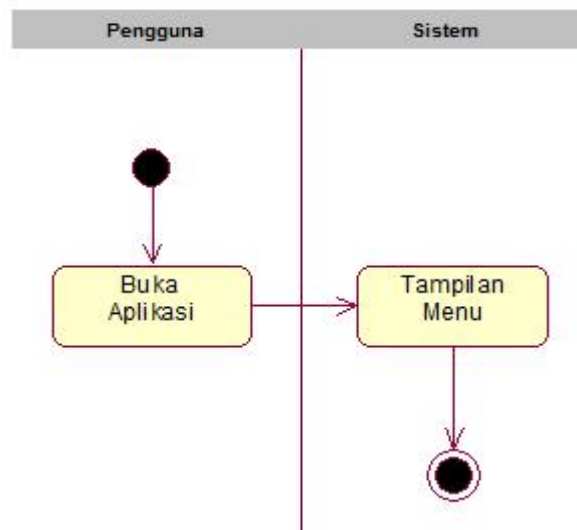
Dari Gambar 4.8. diatas terlihat pengguna memilih menu *Search*. Dimana di dalam menu *Search* terdapat beberapa kategori yaitu ATM, Hotel, Rumah Sakit, Kuliner, SPBU, Transportasi, dan Masjid. Setelah pemilihan kategori selesai proses selanjutnya adalah sistem akan menampilkan *list* lokasi

berdasarkan kategori masing-masing, pilih lokasi yang ingin dituju, lalu sistem akan memanggil koordinat dari database dan melakukan proses pencarian rute dengan menggunakan Algoritma *Floyd-Warshall*. Setelah proses selesai, maka *direction* pencarian dengan rute terpendek akan ditampilkan ke pengguna.

4.5.4. Activity Diagram

Untuk memudahkan dalam perancangan *activity diagram* maka *activity diagram* dalam aplikasi ini akan dipecah menjadi beberapa bagian. Untuk *Activity Diagram* lainnya dapat di lihat pada lampiran A.

A. Activity Diagram Buka Aplikasi



Gambar 4.9. Activity Diagram Buka Aplikasi

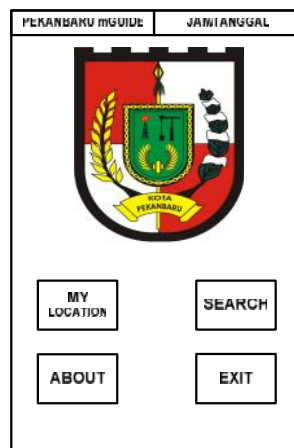
Pada bagian pertama *activity diagram* seperti terlihat pada gambar 4.9 saat aplikasi dijalankan sistem akan memanggil *class* form utama untuk meletakkan aplikasi dalam melakukan inisialisai proses apa saja yang akan di *load* pertama kali, kemudian aplikasi akan menampilkan tampilan menu setelah tampilan menu utama tampil maka proses akan selesai.

4.6. Perancangan Interface

Berikut adalah tampilan interface aplikasi Pekanbaru mGuide yang berjalan pada sistem operasi android. Untuk perancangan *interface* yang lebih lengkapnya dapat dilihat pada Lampiran B.

A. Halaman Utama

Halaman utama pada aplikasi ini tampil apabila aplikasi *Pekanbaru mGuide* dijalankan, tampilan utama berisi menu-menu berupa *My Location*, *Search* dan *About*. tampilan halaman utama pada aplikasi dapat dilihat seperti gambar berikut :



Gambar 4.10. Tampilan Utama *Pekanbaru mGuide*

BAB V

IMPLEMENTASI DAN PENGUJIAN

Bab ini merupakan bagian dari fase *construction* dan fase *transition*, dimana telah dilakukan pengkodean aplikasi, dan akan dilakukan implementasi aplikasi dan pengujian fungsi-fungsi aplikasi dengan metode Blackbox sebagai bagian dari fase *costruction*, dan akan dilakukan pengujian aplikasi terhadap pengguna, kemudian akan dilakukan pengamatan dari hasil pengujian tersebut untuk mengetahui kekurangan aplikasi dan kemudian dilakukan pengambilan kesimpulan sebagai bagian dari fase *transition*.

5.1. Implementasi

Tahapan implementasi merupakan tahapan dimana aplikasi yang telah dirancang, dianalisa, dan dibangun, lalu diuji kelayakannya untuk selanjutnya dioperasikan sebagaimana mestinya sesuai dengan fungsi dan kelayakannya. Berikut ini akan dijelaskan tentang implementasi dari analisis dan perancangan yang telah dilakukan terhadap aplikasi Pekanbaru mGuide berbasis *client-server* pada sistem operasi Android ini.

5.1.2. Lingkungan Pengembangan

Komponen-komponen yang dibutuhkan untuk mengembangkan aplikasi ini antara lain berupa komponen perangkat keras dan perangkat lunak.

1. Perangkat keras

Processor	: <i>Intel(R) Core(TM) i3-2310M CPU @2.20 GHz</i>
Memori (RAM)	: 2 GB
HDD	: 500 GB

2. Perangkat Lunak

Sistem Operasi : *Windows 7 Ultimate 32-bit Operating System*

Bahasa Pemrograman : Java dan PHP

Tools Pengembangan : Eclipse Galileo 3.5, Notepad

. : *Java Development Kit 7u7* (JDK 7u7)

: *Android SDK, ADT 18.0*

: *Android Virtual Device 2.2 (Froyo)*

Browser : *Mozilla Firefox*

Server : XAMPP (Apache 2, MySQL, PhpMyAdmin)

Pemodelan UML : Rational Rose, Microsoft Visio

5.1.3. Lingkungan Implementasi

Untuk lingkungan implementasi aplikasi ini dilakukan dengan menggunakan perangkat keras dan perangkat lunak diantaranya:

1. Perangkat keras : *Smartphone* Android Smartfren Andromax-i
2. Perangkat lunak : Sistem Operasi Android 4.0.3 (ICS)

5.1.4. Tahap-tahap Implementasi

Pada tahap-tahap implementasi ini akan dijelaskan bagaimana tahapan yang penulis lakukan dalam implementasi aplikasi yang telah dibangun. Adapun tahap-tahap yang dilakukan adalah:

a. Instalasi Penghubung

Seperti yang telah dijelaskan pada bab analisa dan perancangan, penghubung mempunyai peran penting untuk aplikasi Pekanbaru mGuide pada sistem operasi Android ini. Penghubung ini berperan sebagai media antara aplikasi di Android dengan *database server*.

Penghubung sendiri dibangun menggunakan bahasa pemrograman PHP, secara umum penghubung ini berisi baris kode yang bekerja meneruskan *query* dan respon dari *client* dan *server*. Algoritma 5.1 di bawah ini merupakan penghubung koneksi yang bertugas sebagai penghubung antara PHP dan *Database Server*. Untuk *source code* lainnya dapat dilihat pada lampiran C.

```

<?php
$link=mysql_connect("localhost", "root", "");
mysql_select_db("fasum");

if(!$link){
die('Could not connect: ' . mysql_error());
}
?>

```

Algoritma 5.1. Koneksi.php

b. Instalasi Aplikasi *Pekanbaru mGuide*

Tahap ini merupakan tahap memasang aplikasi *Pekanbaru mGuide* yang telah dibangun berdasarkan analisa dan perancangan. Aplikasi dipasang pada perangkat Android. Perangkat Android yang digunakan yaitu *smartphone Smartfren Andromax-i*. Untuk melakukan instalasi aplikasi, cukup klik aplikasi *Pekanbaru mGuide* yang telah di *package* ke dalam format *.apk (PKUGuide.apk) pada perangkat Android dan selesai. Aplikasi siap dijalankan.

c. Implementasi Unjuk Kerja Aplikasi Perangkat Android

Implementasi kali ini menggunakan perangkat dengan sistem operasi Android 4.0.4 (Ice Cream Sandwich). Hasil implementasi menampilkan menu *home* gambar 5.1 di bawah ini. Hasil implementasi pada perangkat Android lainnya dapat dilihat pada lampiran D.



Gambar 5.1. Hasil implementasi menampilkan menu *home*

Sistem yang dibangun berjalan dengan baik di perangkat Android, hal ini dilihat dari keberhasilan aplikasi dalam menampilkan menu *Home* di perangkat Android.

5.2. Pengujian *Blackbox* Aplikasi *Pekanbaru mGuide*

Tahapan pengujian dilakukan untuk mengetahui apakah sistem yang dibangun telah sesuai dengan yang di harapkan atau tidak. Pada tahap pengujian sistem ini perangkat keras yang digunakan yaitu *smartphone* Smartfren Andromax-i. Pengujian yang akan dilakukan adalah pengujian akses ke sistem dan pengujian fungsionalitas sistem menggunakan metode *blackbox*. Adapun hasil dari pengujian ini dapat dilihat pada tabel 5.1.



Table 5.1. Pengujian aplikasi dengan metode *Blackbox* (Smartfren Andromax-i)




N O	Nama Pengujian	Deskripsi	Prekondisi	Prosedur Pengujian	Keluaran yang diharapkan	Hasil yang Didapat	Kesimpulan
1.	Menampilkan daftar kategori dan <i>list</i> lokasi	menampilkan kategori dan <i>list</i> lokasi yang ada pada menu <i>search</i>	-	Klik menu <i>search</i> pada perangkat Android	Tampil daftar seluruh kategori dan detail <i>list</i> lokasi berdasarkan kategori yang dipilih	Tampil daftar seluruh kategori dan detail <i>list</i> lokasi berdasarkan kategori yang dipilih	Benar
2.	Menampilkan <i>direction</i> dengan rute terpendek	menampilkan lokasi pengguna dan <i>direction</i> pada menu <i>search</i>	Daftar <i>list</i> lokasi yang dipilih	Klik menu <i>search</i> , pilih salah satu kategori, dan pilih lokasi yang akan dituju	Tampil <i>direction</i> dengan rute terpendek	Tampil <i>direction</i> dengan rute terpendek	Benar
3.	Menampilkan <i>my location</i>	menampilkan beberapa lokasi fasilitas umum	-	Klik menu <i>my location</i> pada perangkat Android	Tampil lokasi pengguna dan lokasi fasilitas umum yang ada	Tampil lokasi pengguna dan lokasi fasilitas umum	Benar
4.	Menampilkan <i>about</i>	Pengujian untuk menampilkan <i>about</i>	-	Klik menu <i>about</i> pada perangkat android	Tampil informasi tentang aplikasi	Tampil informasi tentang aplikasi	Benar




5.3. Pengujian Akses Aplikasi *Pekanbaru mGuide*


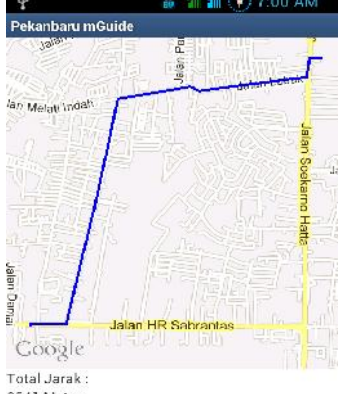
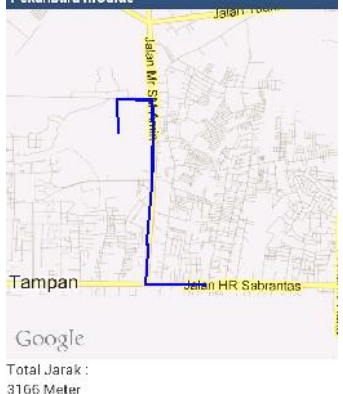
Pengujian yang dilakukan bertujuan untuk mengetahui proses hasil dari sistem, yaitu memperlihatkan aplikasi *Pekanbaru mGuide* yang telah dibangun bisa diakses melalui berbagai perangkat Android dan berbagai versi Android atau tidak. Hasil dari pengujian dapat dilihat halaman *home* telah tampil, dan semua menu serta koordinat yang ditampilkan di sistem sesuai dengan *database server*. Pada tabel 5.2 di bawah ini menjelaskan pengujian akses ke aplikasi *Pekanbaru mGuide* menggunakan beberapa perangkat bersistem operasi Android.




Tabel 5.2. Pengujian akses ke aplikasi dari *device* Android (Smartfren Andromax-i)


Tanggal - Pukul	Lokasi Awal	Lokasi Tujuan	Provider	Jarak Tempuh (Meter)	Waktu Tempuh (Menit)	Tampilan Hasil Pengujian
10 Desember 2012 – 10.30 AM	Jl. Cipta Karya	Aston Pekanbaru Hotel	Smartfren	11.480	17,22	
10 Desember 2012 – 14.00 PM	Jl. Cipta Karya	SPBU Depan Hotel Pangeran	Smartfren	11.248	16,872	

11 Desember 2012 – 14.30 PM	Jl. Cipta Karya	SPBU Simp. Arifin Achmad- Sudirman	Smartfren	9.556	14,3340	 <p>Pekanbaru mGuide</p> <p>Senapelan</p> <p>Sukajadi</p> <p>Sail</p> <p>Simpatiga</p> <p>Google</p> <p>Total Jarak : 9556 Meter Waktu Tempuh (Kecepatan Rata-Rata 40km/h) : 14.333999999999998 Menit</p>
12 Desember 2012 – 10.30 AM	Jl. Cipta Karya	ATM BNI Riau Mandiri	Smartfren	9.184	13,7760	 <p>Pekanbaru mGuide</p> <p>Senapelan</p> <p>Sukajadi</p> <p>Simpatiga</p> <p>Google</p> <p>Total Jarak : 9184 Meter Waktu Tempuh (Kecepatan Rata-Rata 40km/h) : 13.775999999999998 Menit</p>
12 Desember 2012 – 13.00 PM	Jl. Cipta Karya	ATM Mandiri Tambusai	Smartfren	5.828	8,7420	 <p>Pekanbaru mGuide</p> <p>Jalan Tuanku Tambusai</p> <p>Suku</p> <p>Google</p> <p>Total Jarak : 5828 Meter Waktu Tempuh (Kecepatan Rata-Rata 40km/h) : 8.741999999999999 Menit</p>

13 Desember 2012 – 13.30 PM	Jl. Cipta Karya	Masjid Jl. KH. Ahmad Dahlan	Smartfren	9.551	14,3265	 <p>Pekanbaru mGuide</p> <p>Total Jarak : 9551 Meter Waktu Tempuh (Kecepatan Rata-Rata 40km/h) : 14.326500000000001 Menit</p>
14 Desember 2012 – 11.30 AM	Jl. Cipta Karya	CFC Sudirman	Smartfren	11.900	17,850	 <p>Pekanbaru mGuide</p> <p>Total Jarak : 11900 Meter Waktu Tempuh (Kecepatan Rata-Rata 40km/h) : 17.849999999999998 Menit</p>
15 Desember 2012 – 11.00 AM	Jl. Cipta Karya	Hotel Pangeran	Smartfren	10.628	15,942	 <p>Pekanbaru mGuide</p> <p>Total Jarak : 10628 Meter Waktu Tempuh (Kecepatan Rata-Rata 40km/h) : 15.942 Menit</p>

22 Desember 2012 – 13.30 PM	Jl. Cipta Karya	Bandara	Smartfren	8.772	13,158	 <p>Total Jarak : 8772 Meter Waktu Tempuh (Kecepatan Rata-Rata 40km/h) : 13.158 Menit</p>
22 Desember 2012 – 14.30 PM	Jl. Cipta Karya	RS Eka Hospital	Smartfren	3.541	5,3115	 <p>Total Jarak : 3541 Meter Waktu Tempuh (Kecepatan Rata-Rata 40km/h) : 5.3115 Menit</p>
25 Desember 2012 – 14.00 PM	Jl. Cipta Karya	Stadion Utama Riau	Smartfren	3.166	4,749	 <p>Total Jarak : 3166 Meter Waktu Tempuh (Kecepatan Rata-Rata 40km/h) : 4.749 Menit</p>

26 Desember 2012 – 14.30 PM	Jl. Cipta Karya	ATM BCA RS. Syarifa	Smartfren	10.149	15,2235	 <p>Pekanbaru mGuide</p> <p>Total Jarak : 10149 Meter Waktu Tempuh (Kecepatan Rata-Rata 40km/h) : 15.223499999999998 Menit</p>
30 Desember 2012 – 14.30 PM	Jl. Cipta Karya	RS. Awal Bros	Smartfren	10.324	15,486	 <p>Pekanbaru mGuide</p> <p>Total Jarak : 9105 Meter Waktu Tempuh (Kecepatan Rata-Rata 40km/h) : 13.6575 Menit</p>
05 Januari 2012 – 10.30 AM	Jl. Cipta Karya	Hotel Ratu Mayang Garden	Smartfren	8.356	12,534	 <p>Pekanbaru mGuide</p> <p>Total Jarak : 8356 Meter Waktu Tempuh (Kecepatan Rata-Rata 40km/h) : 12.534 Menit</p>

07 Januari 2012 – 14.00 PM	Jl. Cipta Karya	RS Eria Bunda	Smartfren	8.492	12,738	
----------------------------------	--------------------	------------------	-----------	-------	--------	-------------------------------------------------------------------------------------

Tabel 5.3. Pengujian saplikasi *Pekanbaru mGuide* dari beberapa *device* yang berbeda.

Tanggal – Pukul	Perangkat Android	Versi Android	Provider	Pengujian	Hasil
10 Desember 2012 10.30 AM	Sony Ericsson Xperia Mini	Gingerbread (2.3.5)	Telkomsel	Koneksi ke <i>server</i>	Berhasil
				Daftar kategori lokasi	Berhasil
				Daftar <i>list</i> lokasi berdasarkan kategori	Berhasil
				Menampilkan <i>direction</i>	Berhasil
				Menampilkan <i>my location</i>	Berhasil
				Menampilkan <i>about</i>	Berhasil
22 Desember 2012 13.30 PM	Sony Ericsson Xperia Mini	Gingerbread (2.3.5)	3	Koneksi ke <i>server</i>	Berhasil
				Daftar kategori lokasi	Berhasil
				Daftar <i>list</i> lokasi berdasarkan kategori	Berhasil
				Menampilkan <i>direction</i>	Gagal
				Menampilkan <i>my location</i>	Berhasil
				Menampilkan <i>about</i>	Berhasil
23 Desember 2012 09.00 AM	Sony Ericsson Xperia Mini	Gingerbread (2.3.5)	XL	Koneksi ke <i>server</i>	Berhasil
				Daftar kategori lokasi	Berhasil
				Daftar <i>list</i> lokasi berdasarkan kategori	Berhasil
				Menampilkan <i>direction</i>	Gagal
				Menampilkan <i>my location</i>	Berhasil
				Menampilkan <i>about</i>	Berhasil
10 Desember 2012 11.30 AM	Samsung Galaxy Ace Plus	Gingerbread (2.3.6)	Telkomsel	Koneksi ke <i>server</i>	Berhasil
				Daftar kategori lokasi	Berhasil
				Daftar <i>list</i> lokasi berdasarkan kategori	Berhasil
				Menampilkan <i>direction</i>	Berhasil
				Menampilkan <i>my location</i>	Berhasil
				Menampilkan <i>about</i>	Berhasil
22 Desember 2012 14.00 PM	Samsung Galaxy Ace Plus	Gingerbread (2.3.6)	3	Koneksi ke <i>server</i>	Berhasil
				Daftar kategori lokasi	Berhasil
				Daftar <i>list</i> lokasi berdasarkan kategori	Berhasil
				Menampilkan <i>direction</i>	Gagal
				Menampilkan <i>my location</i>	Berhasil
				Menampilkan <i>about</i>	Berhasil

21 Desember 2012 11.30 AM	Samsung Galaxy Ace Plus	Gingerbread (2.3.6)	XL	Koneksi ke <i>server</i>	Berhasil
				Daftar kategori lokasi	Berhasil
				Daftar <i>list</i> lokasi berdasarkan kategori	Berhasil
				Menampilkan <i>direction</i>	Gagal
				Menampilkan <i>my location</i>	Berhasil
				Menampilkan <i>about</i>	Berhasil
11 Desember 2012 11.00 AM	Sony Ericsson Xperia Mini Pro	Ice Cream Sandwich (4.0.4)	Telkomsel	Koneksi ke <i>server</i>	Berhasil
				Daftar kategori lokasi	Berhasil
				Daftar <i>list</i> lokasi berdasarkan kategori	Berhasil
				Menampilkan <i>direction</i>	Berhasil
				Menampilkan <i>my location</i>	Berhasil
				Menampilkan <i>about</i>	Berhasil
11 Desember 2012 11.00 AM	Sony Ericsson Xperia Mini Pro	Ice Cream Sandwich (4.0.4)	3	Koneksi ke <i>server</i>	Berhasil
				Daftar kategori lokasi	Berhasil
				Daftar <i>list</i> lokasi berdasarkan kategori	Berhasil
				Menampilkan <i>direction</i>	Gagal
				Menampilkan <i>my location</i>	Berhasil
				Menampilkan <i>about</i>	Berhasil
22 Desember 2012 09.30 AM	Sony Ericsson Xperia Mini Pro	Ice Cream Sandwich (4.0.4)	XL	Koneksi ke <i>server</i>	Berhasil
				Daftar kategori lokasi	Berhasil
				Daftar <i>list</i> lokasi berdasarkan kategori	Berhasil
				Menampilkan <i>direction</i>	Gagal
				Menampilkan <i>my location</i>	Berhasil
				Menampilkan <i>about</i>	Berhasil
13 Desember 2012 10.30 AM	Smartfren Andromax	Ice Cream Sandwich (4.0.3)	Smartfren	Koneksi ke <i>server</i>	Berhasil
				Daftar kategori lokasi	Berhasil
				Daftar <i>list</i> lokasi berdasarkan kategori	Berhasil
				Menampilkan <i>direction</i>	Berhasil
				Menampilkan <i>my location</i>	Berhasil
				Menampilkan <i>about</i>	Berhasil

5.4. Kesimpulan Pengujian

Setelah dilakukan beberapa pengujian terhadap aplikasi yang telah dibangun, maka dapat ditarik kesimpulan dari hasil pengujian tersebut. Berikut kesimpulannya:

1. Aplikasi *Pekanbaru mGuide* yang dibangun untuk perangkat Android, dapat melakukan koneksi ke *server* dan dapat menampilkan konten sesuai dengan analisa dan perancangan.
2. Aplikasi yang dijalankan di beberapa perangkat Android yang berbeda operator akan mendapatkan hasil yang berbeda pula, seperti pada operator XL dan 3 semua fitur dapat berjalan dengan baik kecuali fitur menu *search*, pada fitur ini ketika aplikasi akan menampilkan *direction* ke pengguna terjadi *force close*. Sedangkan ketika aplikasi berjalan pada operator Smartfren dan Telkomsel fitur menu *search* dapat menghasilkan *direction* dan beberapa fitur menu yang lain.
3. Aplikasi yang dijalankan di versi OS Android yang berbeda tidak menemui kendala *force close*.
4. Kategori, *list* lokasi, dan *direction* yang ditampilkan di perangkat Android semua telah sesuai dengan *database* yang di-*server*.
5. Aplikasi yang dijalankan di perangkat android ini memiliki beberapa kendala, yaitu:
 - a. Kondisi cuaca: kondisi cuaca cerah lebih baik dan cepat GPS untuk nge-*lock* posisinya, dan sebaliknya jika kondisi cuacanya dalam keadaan mendung maka GPS akan mengalami kesulitan untuk nge-*lock* posisinya.
 - b. *Obstacle*/ hambatan, seperti: berada didalam gedung atau ruangan tertutup.
 - c. Kualitas sinyal dari masing-masing operator.
6. Dari pengujian yang dilakukan diatas maka dapat disimpulkan bahwa operator yang terbaik untuk mengakses aplikasi ini adalah Smartfren dan Telkomsel.

7. Waktu yang dibutuhkan dalam mengakses aplikasi jika dalam kondisi terbaik, maksudnya dengan kondisi terbaik adalah dengan cuaca yang cerah, tidak berada di dalam ruangan dan dengan sinyal operator yang baik adalah ± 7 menit.

BAB VI

PENUTUP

6.1 Kesimpulan

Setelah menyelesaikan serangkaian tahapan dalam merancang dan membangun aplikasi *Pekanbaru mGuide* berbasis *client-server* pada sistem operasi Android yang dimulai dari pengumpulan data tentang koordinat jalan, lokasi fasilitas umum, hingga pada tahapan pengujian, maka dapat diambil beberapa kesimpulan diantaranya adalah sebagai berikut:

1. Aplikasi yang dibangun sudah berjalan pada perangkat Android dan bisa mengakses koordinat jalan dan koordinat lokasi yang ada di *server* melalui penghubung (PHP).
2. Aplikasi *Pekanbaru mGuide* sudah bisa menampilkan beberapa kategori fasilitas umum, *list* lokasi, dan *direction* dengan rute terpendek pada perangkat android dalam bentuk *map* menggunakan algoritma *floyd-warshall*.
3. Berbagai perangkat dengan *hardware* dan *software* yang berbeda masih dapat menjalankan semua fitur aplikasi dengan baik.
4. Waktu yang dibutuhkan dalam mengakses aplikasi jika dalam kondisi terbaik, maksudnya dengan kondisi terbaik adalah dengan cuaca yang cerah, tidak berada di dalam ruangan dan dengan sinyal operator yang baik adalah ± 7 menit.

6.2 Saran

Beberapa hal yang disarankan dalam pengembangan aplikasi *Pekanbaru mGuide* berbasis *client server* pada sistem operasi Android ini adalah sebagai berikut:

1. Pada pengembangan aplikasi *Pekanbaru mGuide* selanjutnya diharapkan tidak hanya mampu menampilkan *direction* dengan rute terpendek saja, melainkan juga telah mampu mencari jalur atau rute baru jika ada gangguan yang tidak terduga seperti macet atau banjir.
2. Pada pengembangan aplikasi *Pekanbaru mGuide* selanjutnya diharapkan memiliki lebih banyak fitur lain, seperti: menampilkan informasi dari setiap lokasi dari masing-masing kategori.
3. Pada pengembangan aplikasi *Pekanbaru mGuide* selanjutnya diharapkan dapat dioperasikan tidak hanya di android saja, melainkan juga bisa dioperasikan pada *platform* yang lain.
4. Aplikasi ini memiliki kelemahan, yaitu membutuhkan waktu sekitar 7 menit untuk mendapatkan rute terpendeknya, dan itu terlalu lama prosesnya bagi pengguna. Oleh karena itu, pada pengembangan aplikasi *Pekanbaru mGuide* selanjutnya diharapkan aplikasi dapat melakukan pencarian rute terpendek dengan waktu yang lebih cepat, sebagai contoh dengan cara menggunakan algoritma-algoritma optimasi untuk mempercepat proses pencarian rute tersebut.
5. Pada pengembangan aplikasi *Pekanbaru mGuide* selanjutnya diharapkan aplikasi dapat menampilkan lokasi-lokasi gedung di sekitar lokasi fasilitas umum tersebut. Tujuannya untuk mempermudah pengguna dalam melakukan pencarian lokasi fasilitas umum tersebut.
6. Pada pengembangan aplikasi *Pekanbaru mGuide* selanjutnya diharapkan dapat dibangun aplikasi yang dapat mencari rute terdekat dari kategori yang dipilih oleh pengguna.

DAFTAR PUSTAKA

- Abi. “Global Positioning System” [Online] Available <http://habi3.blogspot.com/2007/05/global-positioning-system-gps.html>, diakses 09 Maret 2012.
- Arsa, Mudi. “Lintasan Terpendek” [Online] Available <http://mudiarsa.blogspot.com/2010/08/lintasan-terpendek.html>, diakses 29 Februari 2012.
- Darwiyanti, Sri dan Romi Satria Wahono. *Pengenalan Unified Modeling Language (UML)* .[Online] Available <http://ilmukoputer.org/2006/08/05/pengantar-uml/> 14 Januari 2012 .
- Fathurahma, “Pemilihan Sekolah Menengah Pertama (SMP) di Pekanbaru Menggunakan Analytical Hierarchy Process (AHP)”. Pekanbaru. 2011.
- Fitriyani, Dian, “Aplikasi LBS (Location Based Services) pada Mobile Phone dengan Teknologi J2ME”. Pekanbaru. 2011.
- Halliday, Resnick, Walker, “Fisika Dasar Edisi 7 Jilid 1”. Erlangga, Surabaya. 2012.
- Jong, Jek S, “Matematika Diskrit dan Aplikasinya pada Ilmu Komputer”. Andi, Yogyakarta. 2006.
- Kamayudi, Apri. “Studi dan Implementasi Algoritma Dijkstra, Bellman-Ford, dan Floyd-Warshall dalam Menangani Masalah Lintasan Terpendek dalam Graf”. Bandung. 2006.
- Nirwan. “Marine Science: Konsep Dasar GIS” [Online] Available <http://marinescience-nirwan.blogspot.com/2011/03/konsep-dasar-gis.html>, diakses 20 November 2011.
- Novandi, Raden A. D. “Perbandingan Algoritma Dijkstra dan Algoritma Floyd-Warshall dalam Penentuan Lintasan Terpendek (Single Pair Shortest Path)”. Strategi Algoritmik, Bandung. 2007.
- Prahasta, Eddy. “Sistem Informasi Geografis: Konsep-Konsep Dasar (Perspektif Geodesi & Geomatika)”. Informatika Bandung, Bandung. 2009.
- Rational Team. “Rational Unified Process : Best Practices for Software Development Teams”. 2001.

- Rizal, Ahmad. “*Sistem Informasi Geografis PDAM*” [Online] Available <http://rizaldicaprio.wordpress.com/my-wife-family-history/>, diakses 18 Desember 2011.
- Safaat, Nazruddin, “*Pemograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*”. Informatika, Bandung. 2011.
- Siswanto. “*Algoritma dan Struktur Data non Linier dengan Java*”. Graha Ilmu, Jakarta. 2010.
- Tanoe, Andre. “*Berkenalan dengan GPS dan Penerapannya pada Kesehatan Masyarakat*”. Pohon Cahaya, Yogyakarta. 2011.
- Widodo, Prabowo. P. “*Menggunakan UML (Unified Modelling Language)*”. Informatika, Bandung. 2011.
- Wiley, John. “*Understanding GIS: The ARC/INFO Method*”. ESRI, California, 1990.